

DIPLOMARBEIT

ACA Tradingbot


Ausgeführt im Schuljahr 2024/25 von:

Karl Pferscher	5BHITM-10
Florian Brunner	5BHITM-01

Betreuer:

Dipl.-Ing. Msc. Paul Panhofer Bsc.
Dipl.-Ing. Msc. Paul Panhofer Bsc.

Krems, am 04.04.2025

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

EIDESSTATTLICHE ERKLÄRUNG


Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Krems, am 04.04 2025

Verfasser/Verfasserinnen:

Karl Pferscher

Florian Brunner

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:


DIPLOMARBEIT

DOKUMENTATION


Namen der Verfasser/innen	Karl Pferscher Florian Brunner
Jahrgang / Klasse Schuljahr	5BHITM 2024/25
Thema der Diplomarbeit	ACA Tradingbot
Kooperationspartner	X-WORKS GmbH

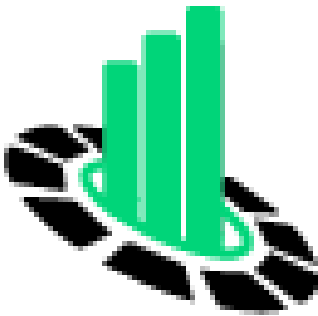
Aufgabenstellung	<p>Einsatz von Apache Spark zur Verarbeitung historischer Marktdaten für einen algorithmischen Tradingbot, basierend auf technischen Indikatoren. Analyse der Verarbeitungsgeschwindigkeit von Apache Spark für einen algorithmischen Tradingbot auf Basis umfangreicher historischer Marktdaten.</p> <p>Untersuchung der Regressionsanalyse anhand analytischer Systeme und technischen Indikatoren zur Auswertung historischer Marktdaten. Effiziente Modellierung großer Datensätze zur Analyse von Zusammenhängen zwischen Indikatoren und Marktentwicklungen liegen im Fokus.</p>
------------------	--

Realisierung	<ul style="list-style-type: none"> • Docker-Container (Apache Spark, Flask API, MySQL-Datenbank, .NET Blazor) • Python-Skripte • Integration von Yahoo Finance API • Kriterienbasierte Bewertung der Verarbeitungsgeschwindigkeit von Apache Spark • Bewertung von Tradingstrategien hinsichtlich der Trefferquote und des ausschlaggebendsten Indikators
--------------	--

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

Ergebnisse	Plattform zur Analyse von Finanzmarktdaten und algorithmischen Handelsentscheidungen über Weboberfläche.
------------	--


	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

Typische Grafik, Foto etc. (mit Erläuterung)	 ACA TRADING BOT
---	--

Teilnahme an Wettbewerben, Auszeichnungen	...
---	-----

Möglichkeiten der Einsichtnahme in die Arbeit	Kein Sperrvermerk
---	-------------------

Approbation (Datum / Unterschrift)	Prüfer/in Paul Panhofer	Abteilungsvorstand / Direktor/in
---------------------------------------	-------------------------	-------------------------------------

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

DIPLOMA THESIS

Documentation

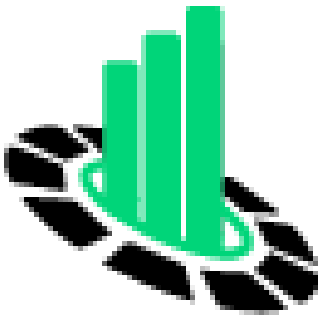
Authors	Karl Pferscher Florian Brunner
Form Academic year	5BHITM 2024/25
Topic	ACA Tradingbot
Co-operation partners	X-WORKS GmbH

Assignment of tasks	<p>Using Apache Spark to process historical market data for an algorithmic trading bot based on technical indicators. Analyzing the processing speed of Apache Spark for an algorithmic trading bot using large historical market data. Investigation of regression analysis using analytical systems and technical indicators to evaluate historical market data. The focus is on efficient modeling of large data sets to analyze correlations between indicators and market developments.</p>
---------------------	--

Realization	<ul style="list-style-type: none"> • Docker containers (Apache Spark, Flask API, MySQL database, .NET Blazor) • Python scripts • Integration of Yahoo Finance API • Criteria-based evaluation of Apache Spark processing performance • Evaluation of trading strategies regarding hit rate and most significant indicator
-------------	--

Results	Platform for financial market data analysis and algorithmic trading decisions via web interface.
---------	--

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

Illustrative graph, photo (incl. explanation)	 ACA TRADING BOT
---	--

Participation in competitions Awards	...
---	-----

Accessibility of diploma thesis	no blocking notice
---------------------------------	--------------------


Approval (Date / Sign)	Examiner Paul Panhofer	Head of Department / / College
---------------------------	------------------------	-----------------------------------

Inhaltsverzeichnis

1. Präambel	11
1.1. Zusammenfassung	11
1.2. Abstract	11
1.3. Danksagung	12
1.4. Gendererklärung	13
1.5. Team	13
2. Einleitung	14
2.1. Ausgangslage und Motivation der Arbeit	14
2.1.1. Bedeutung von Big Data in der Finanzanalyse	14
2.1.2. Herausforderungen im algorithmischen Trading	14
2.2. Forschungsfragen und Zielsetzung	15
2.2.1. Bewertung der Verarbeitungsgeschwindigkeit von Apache Spark bei der Verarbeitung historischer Marktdaten für einen algorithmischen Tradingbot	15
2.2.2. Regressionsanalyse zur Analyse historischer Marktdaten	15
2.3. Themenstellung	16
2.3.1. Big Data und algorithmisches Trading in der Finanzindustrie	16
2.3.2. Relevanz der Regressionsanalyse in der Finanzanalyse	16
2.4. Aufbau und Methodik der Arbeit	16
2.4.1. Methodik der Strategiebewertung	16
2.4.2. Methodik zur Messung der Verarbeitungsgeschwindigkeit von Apache Spark	17
3. Theoretische Grundlagen	19
3.1. Chart-Daten-Big Data und verteilte Datenverarbeitung	19
3.1.1. Charakteristika von Finanz-Big Data	19
3.1.2. Verteilte Datenverarbeitung	19
3.2. Algorithmisches Trading und technische Indikatoren	20
3.2.1. Prinzipien des algorithmischen Tradings	20
3.2.2. Einführung in technische Indikatoren	21
3.3. Regressionsanalyse in der Finanzanalyse	31
3.3.1. Regressionsmodelle und ihre Bedeutung	31
3.3.2. Ansätze und Anwendungen der Regressionsanalyse	32
3.4. Anforderungen an eine skalierbare Datenverarbeitungsarchitektur	33
3.4.1. Anforderungen an den Datenfluss für Finanzdaten	33
3.4.2. Integrationsanforderungen für technische und analytische Systeme . .	33
3.5. Funktionsweise und Architektur von Apache Spark	35
3.5.1. Grundkonzepte und Datenverarbeitung in Spark	35
3.5.2. Eignung und Vorteile für die Finanzdatenverarbeitung	37
3.5.3. Vergleich von Apache Spark mit alternativen Big-Data-Technologien für die Finanzdatenverarbeitung	38
4. Datenverarbeitungskonzepte in Apache Spark	40
4.1. Implementierung der technischen Indikatoren in Spark	40
4.1.1. Auswahl und Implementierung relevanter technischer Indikatoren . .	40

4.2.	Implementierung der analytischen Methodiken	45
4.2.1.	Implementierung des Regressionsmodells	45
4.3.	Implementierung der Datenverarbeitung für historische Marktdaten	47
4.3.1.	Erstmalige Speicherung und Transformation von Finanzmarktdaten in Apache Spark	47
4.3.2.	Kontinuierliche Aktualisierung der Finanzmarktdaten in Apache Spark	52
5.	Bewertung der Untersuchungsergebnisse	58
5.1.	Bewertung der Spark Verarbeitungsgeschwindigkeit	58
5.1.1.	Testaufbau und Rahmenbedingungen	58
5.1.2.	Messmethodik	59
5.1.3.	Ergebnisse und Kennzahlen	59
5.2.	Analyse der Implementierten Strategien und deren Aufbau	61
5.2.1.	Vergleich der unterschiedlichen Strategieimplementierungen	61
5.2.2.	Optimierungsmöglichkeiten der Strategien für große Datenmengen . .	66
6.	Leistungsevaluation und Benchmarking	67
6.1.	Analyse der Verarbeitungsgeschwindigkeit von Apache Spark	67
6.1.1.	Ergebnisse der Messungen: Verarbeitungsgeschwindigkeit von Apache Spark bei historischen Marktdaten	67
6.2.	Effizienz und Genauigkeit der Strategien	69
6.2.1.	Vergleich der Strategien hinsichtlich Trefferquote(Hit-Rate)	69
6.2.2.	Vergleich der durch die Regressionsanalyse ermittelte ausschlaggebende Indikator	70
7.	Prototyp Architektur	73
7.1.	Systemarchitektur	73
7.1.1.	Kommunikationsprotokolle und Systemarchitektur	73
7.1.2.	Prozesskommunikation	74
7.1.3.	Verwendete Libraries zur Datenverarbeitung	75
7.2.	Relationale Datenbank im finanzspezifischen Kontext	77
7.2.1.	Speicheranforderungen in MySQL	77
7.2.2.	Datenkonsistenz und -sicherheit	77
8.	Fazit und Ausblick	78
8.1.	Zusammenfassung der Ergebnisse im Bezug auf Datenverarbeitung und Regressionsmodelle	78
8.1.1.	Schlussfolgerungen aus den Performance-Tests bei der Datenverarbeitung	78
8.1.2.	Relevanz der Regressionsanalyse für Finanzvorhersagen	78
9.	Projektdokumentation	79
9.1.	Arbeitsteilung	79
9.2.	Stundenliste	79
9.2.1.	Florian Brunner:	79
9.2.2.	Karl Pferscher:	83
I.	Literaturverzeichnis	87

II. Abbildungsverzeichnis	90
III. Tabellenverzeichnis	91
IV. List of Listings	92
V. Abkürzungsverzeichnis	93
A. Appendix	94
A.1. Kapitelverzeichnis	94
A.2. Besprechungsprotokolle	96
A.3. Pflichtenheft	103
A.4. User Interface	107
A.5. Datenträgerbeschreibung und Installationsanleitung	117
A.5.1. Datenträgerbeschreibung	117
A.5.2. Installationsanleitung	117
A.6. Arbeitspaketbeschreibung und PSP	119

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

1. Präambel

1.1. Zusammenfassung


Im Rahmen dieser Diplomarbeit wurde ein Tradingbot entwickelt, der mehrere Depots verwalten und Kauf- und Verkaufsentscheidungen treffen kann. Die Anwendung umfasst eine benutzerfreundliche Weboberfläche sowie eine robuste Backend-Architektur zur Analyse, Speicherung und Verarbeitung von Finanzmarktdaten.

- Blazor Web App
- MySQL Datenbank
- Flask API

1.2. Abstract

As part of this diploma thesis, a trading bot was developed that can manage multiple portfolios and make buy and sell decisions. The application includes a user-friendly web interface and a robust backend architecture for analyzing, storing, and processing financial market data.

- Blazor Web App
- MySQL Datenbank
- Flask API


	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

1.3. Danksagung

Unser besonderer Dank gilt Herrn Dipl.-Ing. Paul Panhofer für seine engagierte Betreuung unserer Diplomarbeit. Mit seinem technischen Know-how, seinen hilfreichen Rückmeldungen und seiner ständigen Bereitschaft zur Unterstützung hat er entscheidend zum Gelingen dieses Projekts beigetragen. Durch seine strukturierte Herangehensweise und motivierende Art fühlten wir uns stets gut begleitet.

Ebenso möchten wir unseren Mitschülern und Freunden danken, die uns durch ihre Ratschläge, ihren Zuspruch und ihre Unterstützung in anspruchsvollen Phasen motiviert haben.

Ein großer Dank gilt auch unseren Familien, die uns während des gesamten Schuljahres den Rücken gestärkt, uns ermutigt und uns in jeder Hinsicht unterstützt haben.

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

1.4. Gendererklärung

Zur besseren Lesbarkeit der Diplomarbeit wurde ausschließlich die männliche Form verwendet. Da Begriffe wie "Benutzerinnen und Benutzer" den Text unleserlich machen, wurde es schlicht auf "Benutzer" gekürzt, dies soll jedoch keine Geschlechterdiskriminierung zum Ausdruck bringen.

1.5. Team

Das Projektteam besteht aus dem Projektleiter Karl Pferscher und Kollegen Florian Brunner. Betreuer ist Dipl.-Ing. Paul Panhofer Bsc, der Auftraggeber ist DI Mag. Georg Ungerböck.

2. Einleitung

2.1. Ausgangslage und Motivation der Arbeit

2.1.1. Bedeutung von Big Data in der Finanzanalyse

Der Begriff ‘Big Data‘ bezeichnet Datenmengen, welche zu komplex oder zu groß sind, um herkömmlich verarbeitet zu werden. Diese Daten werden aus unterschiedlichen Quellen wie Transaktionen oder sozialen Medien aggregiert. (1)

Im Zuge der zunehmenden Popularität von Big Data in den letzten Jahren hat sich dessen Einfluss auf die Finanzbranche erheblich verstärkt. (2) Durch den Einsatz moderner Technologien können umfangreiche Datenmengen erfasst und systematisch für analytische Zwecke genutzt werden. Die Analyse von Markttrends ermöglicht es zukünftige Marktentwicklungen vorherzusagen. (3)

Anwendungsbereiche:

- Präzisere Entscheidungsfindung
- Effektiveres Risikomanagement
- Verbesserte Kundenanalyse
- Prognose von Markttrends und -entwicklungen

(3)

2.1.2. Herausforderungen im algorithmischen Trading

Der algorithmische Handel stellt einen hochkomplexen und technologisch fortschrittlichen Bereich dar, der eine zentrale Rolle im modernen Aktienmarkt einnimmt. Dieser Ansatz basiert auf der Anwendung spezialisierter Algorithmen und Handelsstrategien, die automatisiert durch Computerprogramme Handelsentscheidungen treffen und Transaktionen ausführen. Der automatische Handelsprozess eröffnet sowohl vielversprechende Chancen als auch potenzielle Risiken.(4)

Vorteile	Nachteile
Komplexe Handelsstrategien	Algorithmische Fehler
Automatisierung von Prozessen	Abhängigkeit von Daten
24/7-Verfügbarkeit	Überreaktion auf Marktereignisse
Reduzierung von Emotionen	Datenlecks und Cyberangriffen

Tabelle 2.1.: Vor und Nachteile des algorithmischen Tradings (4; 5)

2.2. Forschungsfragen und Zielsetzung

2.2.1. Bewertung der Verarbeitungsgeschwindigkeit von Apache Spark bei der Verarbeitung historischer Marktdaten für einen algorithmischen Tradingbot

Untersuchungsanliegen

Ziel dieser Untersuchung ist es, zu analysieren, wie sich die Verarbeitungsgeschwindigkeit von Apache Spark verändert, wenn die Menge historischer Marktdaten wächst. Es wird geprüft, ob die Verarbeitungszeit linear mit der Datenmenge ansteigt oder ob Spark größere Datenmengen effizienter bewältigen kann.

Dabei werden verschiedene Datengrößen getestet und die Verarbeitungszeiten gemessen. Besondere Aufmerksamkeit gilt der Frage, ob Apache Spark durch seine verteilte Architektur bei zunehmender Last eine stabile Performance aufrechterhält oder ob Engpässe auftreten.

Geplantes Ergebnis

Erwartet wird eine detaillierte Analyse, wie Apache Spark auf verschiedene Datenmengen reagiert. Die Untersuchung soll zeigen, ob Spark große historische Datensätze effizient verarbeitet oder ob die Verarbeitungszeit überproportional ansteigt.

2.2.2. Regressionsanalyse zur Analyse historischer Marktdaten

Untersuchungsanliegen


Wie hängen technische Indikatoren mit der Marktentwicklung zusammen

Die Regressionsanalyse untersucht historische Marktdaten, um Beziehungen zwischen technischen Indikatoren und Marktentwicklungen zu identifizieren. Ziel ist es, Abhängigkeiten zu modellieren, die als Grundlage für Handelsentscheidungen dienen. Analytische Systeme verarbeiten große Datenmengen, um Muster und Trends sichtbar zu machen.

Der Fokus liegt auf Indikatoren wie gleitenden Durchschnitten, Volatilitätsmaßen und Momentum-Indikatoren, die Kursveränderungen und Handelsvolumina beeinflussen. Die Modellierung dieser Beziehungen ermöglicht eine datengestützte Entscheidungsfindung für den Aktienmarkt.

Geplantes Ergebnis

Ziel ist die Entwicklung einer Handelsstrategie, welche historische Marktdaten und technische Indikatoren analysiert, und mittels der Regressionsanalyse den aussagekräftigsten Indikator dieser Strategie zu identifizieren.

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

2.3. Themenstellung

2.3.1. Big Data und algorithmisches Trading in der Finanzindustrie

Durch den Einsatz des Algorithmischen Tradings in der Big Data Ära lassen sich große Datenmengen sehr effizient verarbeiten. Dem zufolge ist es möglich Signale oder Trendmuster mit einer komplexen Struktur sehr schnell zu erkennen und aus diesen eine Anlageentscheidung zu treffen. Strategien lassen sich somit flexibel an den Markt anpassen.(6)

2.3.2. Relevanz der Regressionsanalyse in der Finanzanalyse

Die Regressionsanalyse ist ein Analyse Instrument mit dem der Zusammenhang zwischen zwei oder mehreren Variablen ermittelt wird. Die Lineare Regression wird in der Finanz Branche verwendet, um den Zusammenhang zwischen, aus Finanzdaten berechneten, mathematischen Werten und der Marktentwicklung zu eruieren. Anhand dieses Zusammenhangs können Vorhersagen im Finanzmarkt getroffen werden.(7)

2.4. Aufbau und Methodik der Arbeit

2.4.1. Methodik der Strategiebewertung

Im Folgenden wird die Methodik zur Bewertung der algorithmischen Trading-Strategien etabliert. Ziel ist es, anhand historischer Marktdaten die Performance der Strategien zu evaluieren und den maßgeblichen Korrelationskoeffizienten identifizieren.


Implementierung der Trading-Strategien und des Regressionsmodells

Zunächst werden die Trading-Strategien implementiert, die auf verschiedenen technischen Indikatoren basieren. Diese Indikatoren generieren jeweils unterschiedliche Kauf- und Verkaufssignale. Ein Trade wird dabei als die Kombination eines Kaufsignals mit einem darauffolgenden Verkaufssignal definiert. Sollte für eine Aktie lediglich ein Kaufsignal auftreten, erfolgt der Verkauf automatisch am letzten verfügbaren Datum zum jeweils aktuellen Preis.

Ergänzend zur Strategie-Implementierung wird ein multiples lineares Regressionsmodell aufgebaut. In diesem Modell dienen die unterschiedlichen Indikatoren als unabhängige Variablen, während ein Erfolgskriterium der Strategie (Profit) als abhängige Variable herangezogen wird. Ziel dieser Regressionsanalyse ist es, den ausschlaggebendsten Indikator zu identifizieren, der maßgeblich die Performance der Strategie beeinflusst.

Datengrundlage und Testumgebung

Für die Bewertung der Strategien werden historische Marktdaten von 12 Aktien herangezogen, die aufgrund ihrer Popularität ausgewählt wurden. Diese Daten bieten einen vielfältigen Einblick in verschiedene Marktsegmente und -dynamiken. Zur Erfassung unterschiedlicher zeitlicher Perspektiven werden folgende Zeiträume berücksichtigt:

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

1-Stunden-Daten: Ab Juni 2024 Tagesdaten: Ab Dezember 2023 Wochendaten: Ab Dezember 2022 Monatsdaten: Ab März 2020

Die gesammelten Daten werden in ein einheitliches Format überführt und entsprechend aufbereitet, um eine konsistente und vergleichbare Testumgebung für das Backtesting zu schaffen.

Backtesting und Analytische Bewertung

Im nächsten Schritt erfolgt das Backtesting der implementierten Trading-Strategien auf Basis der vorbereiteten historischen Daten. Dabei werden sämtliche generierten Trades, also die Kombinationen von Kauf- und Verkaufssignalen, systematisch erfasst und analysiert. Zur quantitativen Bewertung der Strategie-Performance wird insbesondere die Hit-Rate¹ herangezogen. Diese misst den Anteil der erfolgreichen Trades an der Gesamtzahl der ausgeführten Trades und wird über alle 12 Aktien hinweg ermittelt.

2.4.2. Methodik zur Messung der Verarbeitungsgeschwindigkeit von Apache Spark

Ziel dieser Untersuchung ist es, die Veränderung der Verarbeitungsgeschwindigkeit von Apache Spark bei wachsender Datenmenge zu analysieren. Es soll festgestellt werden, ob die Verarbeitungszeit proportional beziehungsweise linear mit der Datenmenge ansteigt oder ob Spark größere Datenmengen effizienter verarbeiten kann. Dafür wird ein experimenteller Ansatz gewählt, bei dem unterschiedliche Datengrößen getestet und die jeweiligen Verarbeitungszeiten gemessen werden.

Testdatengrundlage


Für die Untersuchung werden historische Marktdaten von mindestens 30 Wertpapieren verwendet. Die Daten stammen aus der Yahoo Finance API und werden mit einem Python-Skript in regelmäßigen Zeitintervallen abgerufen. Um eine möglichst realitätsnahe Umgebung zu simulieren, werden unterschiedliche Zeitspannen (1 Stunde, 1 Tag, 1 Woche, 1 Monat) für die Datenextraktion berücksichtigt.

Testumgebung und Implementierung

Die Tests werden in einer verteilten Umgebung durchgeführt, wobei Apache Spark innerhalb eines Docker-Containers läuft. Die Datenverarbeitung erfolgt durch eine Spark-Applikation, die die abgerufenen Finanzdaten verarbeitet und in eine MySQL-Datenbank speichert. Dabei werden verschiedene Datenmengen analysiert, um zu messen, wie sich die Performance von Spark bei wachsender Last verändert.

Messverfahren

¹Trefferquote

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

Für jede getestete Datenmenge wird die Verarbeitungszeit von Spark gemessen. Die folgenden Faktoren werden dokumentiert:

- Gesamte Verarbeitungszeit (vom Laden der Daten bis zur Speicherung in der Datenbank)
- CPU- und Speicher-Auslastung während der Verarbeitung
- Veränderung der Verarbeitungszeit bei Verdopplung oder Vervielfachung der Datenmenge

Diese Messungen sollen eine detaillierte Analyse ermöglichen, ob die Performance von Apache Spark stabil bleibt oder ob Engpässe auftreten, die Optimierungsmaßnahmen erfordern.

Erwartete Ergebnisse

Die Untersuchung soll aufzeigen, ob Apache Spark bei steigender Datenmenge eine konstante Skalierbarkeit bietet oder ob die Verarbeitungszeit überproportional ansteigt.

3. Theoretische Grundlagen

3.1. Chart-Daten-Big Data und verteilte Datenverarbeitung

3.1.1. Charakteristika von Finanz-Big Data

Finanz-Big Data beschreibt die riesigen Mengen an strukturierten und unstrukturierten Daten, die aus Finanzmärkten generiert werden. Diese Daten umfassen historische Marktdaten, Echtzeit-Tick-Daten, Orderbuch-Daten, Unternehmensberichte, makroökonomische Indikatoren und alternative Datenquellen wie Nachrichten oder soziale Medien.

Ein zentrales Kennzeichen von Finanz-Big Data ist die enorme Geschwindigkeit (Velocity), mit der Daten erzeugt und verarbeitet werden müssen. Finanzmärkte handeln mittlerweile im Mikrosekundenbereich, was zu kontinuierlich wachsenden Datenströmen führt. Gleichzeitig stellt auch das immense Volumen (Volume) der Daten eine Herausforderung dar, da die Mengen oft im Terabyte- oder Petabyte-Bereich liegen, insbesondere bei der langfristigen Speicherung historischer Tick-Daten. Zudem spielt die Vielseitigkeit (Variety) der Daten eine wichtige Rolle: Neben strukturierten Kursdaten existieren ebenso semi-strukturierte Informationen wie Unternehmensberichte und unstrukturierte Inhalte aus Nachrichtenquellen. Gerade diese Vielfalt stellt traditionelle relationale Datenbanken vor erhebliche Schwierigkeiten, da diese Systeme primär auf strukturierte Daten ausgelegt sind und mit der Verarbeitung von semi-strukturierten und unstrukturierten Informationen schnell an ihre Grenzen stoßen. (8)

Die Analyse dieser komplexen und vielfältigen Daten erfordert daher spezialisierte technologische Lösungen. Finanz-Big Data setzt vermehrt auf verteilte Datenbanken oder Big-Data-Frameworks wie Apache Spark, um sowohl Geschwindigkeit, Volumen als auch Vielseitigkeit effizient zu bewältigen.

Relevanz für algorithmisches Trading

Im algorithmischen Trading spielen Finanz-Big Data eine zentrale Rolle. Die Genauigkeit von Handelsmodellen hängt von der Qualität und Quantität der verwendeten Daten ab. Durch die Analyse historischer Daten können Muster und Trends identifiziert werden, die als Grundlage für algorithmische Handelsstrategien dienen.

3.1.2. Verteilte Datenverarbeitung

Verteilte Datenverarbeitung bezeichnet die Aufteilung von Datenverarbeitungsaufgaben auf mehrere miteinander verbundene Computer oder Knoten innerhalb eines Netzwerks. Im Gegensatz zur zentralisierten Datenverarbeitung, bei der alle Prozesse auf einem einzigen System ablaufen, ermöglicht die verteilte Architektur die parallele Bearbeitung von Aufgaben, was zu einer erhöhten Effizienz und Skalierbarkeit führt. (9)

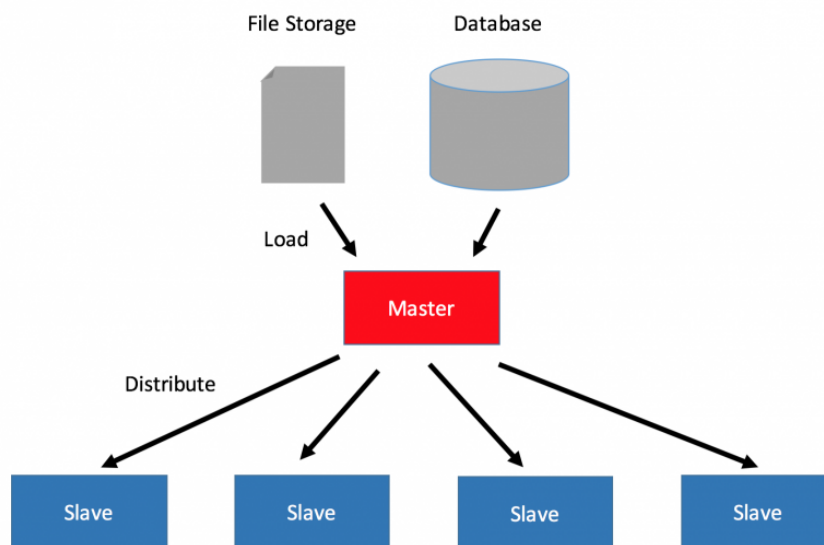


Abbildung 3.1.: Verteilte Datenverarbeitung

Vorteile der verteilten Datenverarbeitung: (10)

- **Skalierbarkeit:** Durch das Hinzufügen weiterer Knoten kann die Rechenkapazität entsprechend dem wachsenden Datenvolumen erweitert werden.
- **Fehlertoleranz:** Fällt ein Knoten aus, können andere Knoten dessen Aufgaben übernehmen, wodurch die Gesamtzuverlässigkeit des Systems gesteigert wird.
- **Geografische Verteilung:** Daten und Prozesse können über verschiedene Standorte verteilt werden, was die Effizienz in globalen Anwendungen erhöht.
- **Performance:** Parallelverarbeitung, als zentrales Merkmal verteilter Datenverarbeitung, steigert die Leistung erheblich. Indem komplexe Aufgaben in kleinere Teilaufgaben zerlegt und auf mehrere Knoten verteilt werden, lassen sich Daten deutlich schneller und effizienter verarbeiten.


Anwendung in der Finanzbranche:

In der Finanzindustrie ist die verteilte Datenverarbeitung besonders relevant. Finanzinstitute verarbeiten täglich enorme Mengen an Transaktionsdaten, Marktinformationen und Kundenprofilen. Durch den Einsatz verteilter Systeme können diese Daten effizient analysiert werden, um beispielsweise Echtzeit-Transaktionen zu ermöglichen, Risiken zu bewerten oder personalisierte Finanzdienstleistungen anzubieten. (10)

3.2. Algorithmisches Trading und technische Indikatoren

3.2.1. Prinzipien des algorithmischen Tradings

Algorithmisches Trading beschreibt die Entscheidungsfindung im Aktienmarkt auf Grundlage von ausgeführtem Code, welcher zur Evaluierung des Marktes, finanzmathematische Formeln nutzt. Eine auf Regeln basierte Strategie baut auf den berechneten Größen auf. (11)

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

Verwendete Variablen:

- Eröffnungskurs
- Schlusskurs
- Handelsvolumen

(11)


Trading-algorithmen werden im Normalfall unter menschlicher Aufsicht genutzt. Der Algorithmus gibt eine Entscheidung vor und der Mensch hat die Wahl über Kauf oder Verkauf. Durch die richtige Kombination und Optimierung der Parameter soll eine effektive Strategie entwickelt werden.(11)

Das Backtesting der Trading-Strategie erfolgt über ein Backtesting-Modul, welches den Algorithmus mit historischen Marktdaten testet und die Trefferquote berechnet. Backtesting sollte nicht als Erfolgsgarantie gesehen werden. (11)

3.2.2. Einführung in technische Indikatoren

Technische Indikatoren sind finanzmathematische Berechnungen, welche auf dem Preis eines Wertpapiers und dem Handelsvolumen basieren. Sie werden in der Finanzbranche genutzt die Untersuchung statischer Informationen Richtung und Trend eines Wertpapiers vorherzusagen.(12)

Die Kombination der Indikatoren als auch die Nutzung einzelner Indikatoren kann verwendet werden, um Handelssignale zu generieren oder die Stärke eines Trends zu bestimmen.(12)

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

Arten der Technischen Indikatoren:

- **Trendindikatoren**
 - Verwendung in der Trendbestimmung
- **Momentum Indikatoren**
 - Messung der Geschwindigkeit bezogen auf Preisbewegungen
 - Signalisierung möglicher Trendumkehr
- **Volumenindikatoren**
 - Messung des Handelsvolumens
 - Signalisiert mögliche Preisbewegungen
- **Volatilitätsindikatoren**
 - Messung der Schwankungsbreite der Preisbewegungen

(12)

Die folgenden Indikatoren wurden für die Strategie aufgrund der Theorie der self-fulfilling-Prophecy ausgewählt. (13)

Trendindikatoren

SMA (Simple Moving Average)

Gleitende Durchschnitte werden regelmäßig angewandt, um Preisfluktuationen (Volatilitäten) zu glätten, und Trends einfacher zu erkennen. Eher in deutlich tendierenden Märkten gut anwendbar. Schlechter anwendbar in Seitwärtstrends. (14)

Der SMA fungiert als Widerstand und Unterstützungslinie oder zur Erkennung einer übergeordneten Trendrichtung. Die Kursbewegungen werden nicht in Realtime wiedergegeben. (14)

Berechnung:

$$\text{SMA} = (\text{Summe der Schlusskurse über den Zeitraum}) / (\text{Anzahl der Perioden}) \quad (15)$$



Abbildung 3.2.: SMA (50, 100, 200) Apple, 4h

SMA-Regeln

- **Trendbestimmung**
 - Abwärtstrend: Kurs unterhalb eines fallenden SMA
 - Aufwärtstrend: Kurs oberhalb eines steigenden SMA
- **Handelssignale**
 - Kaufsignal: Kurs schneidet SMA von unten nach oben (Aufwärtstrend)
 - Verkaufssignal: Kurs schneidet SMA von oben nach unten (Abwärtstrend)
- **Mehrere SMAs kombinieren**
 - Kreuzt ein kürzerer SMA (z. B. 50 Tage) einen längeren SMA (z. B. 100 Tage) und einen SMA (z. B. 200 Tage), kann dies einen Trendwechsel anzeigen
 - Diese Signale werden Golden-Cross und Death-Cross genannt

(14)

EMA (Exponentieller Gleitender Durchschnitt)

Die Periode des EMAs ist frei wählbar jedoch sind bei höherer Perioden Anzahl weniger Fehlsignale enthalten, da das EMA auf kleine Kursschwankungen reagiert.(16)

Berechnung:

Variablen:

- Smoothing Factor (SF)
– $SF = 2 / (n + 1)$
- Schlusskurs (Ct)
- EMA der Vorperiode (EMAt-1)

(16)

Formel: $EMAt = Ct * SF + (1 - SF) * EMAt-1$ (16)



Abbildung 3.3.: EMA (50) Apple, 4h

EMA-Regeln -> siehe SMA-Regeln

Bei der Verwendung eines EMA sollte ein zweiter Indikator (zb: SMA) verwendet werden, da das EMA jüngere Signale mehr gewichtet und sonst Fehlsignale erzeugt werden. Die Überschneidung der Graphen ist signalgebend. (16)

MomentumIndikatoren

RSI (Relative Strength Index)

Der RSI wird verwendet, um zu erkennen, ob ein Wertpapier überkauft oder überverkauft ist. Dies wird durch den Vergleich von durchschnittlichen Kursanstiegen mit durchschnittlichen Kursverlusten erreicht. Aufgrund seiner Sensitivität und schnellen Reaktion kann der RSI Ein- und Ausstiegssignale generieren.(17)

Dadurch, dass er ein geglätteter Oszillator ist, werden ausreißende Signale bei der Signalgebung nicht berücksichtigt. Der RSI gilt in einer Zeitperiode von 14 Tagen/Wochen/Monaten oder auch von 7, 9, 25 Tagen anzuwenden. Als Ergebnis der Berechnung wird ein Wert zwischen 0 und 100 erwartet.(17)



Abbildung 3.4.: RSI Apple, 4h

Berechnung:

Variablen:

- Summe(hoch)
 - Der Mittelwert aller positiven Kursveränderungen über den Betrachtungszeitraum
- Summe(tief)
 - Der Mittelwert aller negativen Kursveränderungen über den Betrachtungszeitraum

(17)

Formel:

$$RSI = \frac{\text{Summe(hoch)}}{\text{Summe(hoch)} + \text{Summe(tief)}}$$

(17)

RSI-Regeln

- Überkaufte und überverkaufte Zonen
 - Über 70: Überkauft → Verkaufssignal
 - Unter 30: Überverkauft → Kaufsignal

(17)

Stochastik

Schlusskurse werden mit einer Kursspanne über eine bestimmte Periode verglichen, um überkaufte und überverkaufte Signale zu identifizieren. Der Stochastik-Indikator eignet sich gut für Seitwärts-Trends und schlecht tendierende Märkte. Er erweist sich als besonders effektiv bei der Anwendung in dynamischen und schnelllebigen Märkten.(18)

Komponenten:

- %K-Linie
 - Höchstes Hoch und Tiefstes Tief werden über eine Zeitspanne verglichen
 - Daraus wird eine Kursspanne definiert
 - Aktueller Schlusskurs wird als Prozentsatz der Kurspanne angegeben
- %D-Linie
 - Gleitender Durchschnitt von %K

(18)

Berechnung:

Variablen:

- C
 - Der aktuelle Schlusskurs (Close)
- Ln
 - Der niedrigste Schlusskurs innerhalb der betrachteten Zeitperiode n
- Hn
 - Der höchste Schlusskurs innerhalb der betrachteten Zeitperiode n

(18)

Formel:

$$\%K = \frac{C - L_n}{H_n - L_n}$$

(18)



Abbildung 3.5.: Stochastik Apple, 4h

Stochastik-Regeln

- Überkaufte und überverkaufte Zonen
 - Verkaufssignal (>80): Die %K-Linie kreuzt die %D-Linie von oben nach unten. Das Verlassen der 80er-Zone nach unten signalisiert das Ende eines Aufwärtstrends.
 - Kaufsignal (<20): Die %K-Linie kreuzt die %D-Linie von unten nach oben. Das Verlassen der 20er-Zone nach oben signalisiert das Ende eines Abwärtstrends.

(18)

Momentum

Der Momentum-Indikator erfasst die Dynamik und Intensität von Kursbewegungen. Er dient dazu, die "Schwungkraft" der Preise zu analysieren und Hinweise auf Änderungen in deren Richtung oder Tempo zu geben. (19)

Berechnung:

Variablen:

- C_t
 - Der aktuelle Schlusskurs
- C_{tn}
 - Der Schlusskurs von vor n Tagen

(19)

Formel:

$$\text{Momentum} = C_t - C_{tn}$$

(19)



Abbildung 3.6.: Momentum Apple, 4h

Momentum-Regeln

- Momentum > 0
 - Steigend: Zunehmende positive Schwungkraft → Fortsetzung des Aufwärtstrends
 - Fallend: Abnehmende Schwungkraft → Möglicher Beginn einer Abschwächung oder Trendumkehr
- Momentum < 0
 - Steigend: Abnehmende Schwungkraft → Möglicher Beginn einer Erholung oder Trendumkehr
 - Fallend: Zunehmende negative Schwungkraft → Fortsetzung des Abwärtstrends

(19)


Volatilitätsindikatoren

Bollinger Bänder

Die Bollinger Bänder werden eingesetzt um die Ober- und Untergrenze einer Preisbewegung hervorzuheben. Durch diesen Indikator können Widerstands und Unterstützungszonen gekennzeichnet werden, wenn sich kein klarer Trend herauskristallisiert.(20)



Abbildung 3.7.: Bollinger Bänder Apple, 4h

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

Komponenten:

- Oberes und Unteres Band
 - Jeweils der obere und untere Abstand von zwei Standardabweichungen eines 20 SMAs
 - 95% der Preise sollten innerhalb der Bänder liegen

(20)

Bollinger Bänder-Regeln

- Obereres Band
 - Eine Berührung des oberen Bandes ist ein Indiz für einen potentiellen Abwärtstrend.
- Unteres Band
 - Eine Berührung des unteren Bandes ist ein Indiz für einen potentiellen Aufwärtstrend.

(20)

3.3. Regressionsanalyse in der Finanzanalyse

3.3.1. Regressionsmodelle und ihre Bedeutung

Lineare Regression

Bei einer linearen Regression wird eine abhängige Variable durch eine oder mehrere unabhängige Variablen ermittelt. Durch ein Lineares Modell (LM) können nur die Zusammenhänge zugezogen werden, bei welchen die abhängige Variable eine Linearkombination der jeweiligen Regressionskoeffizienten ist. (21)

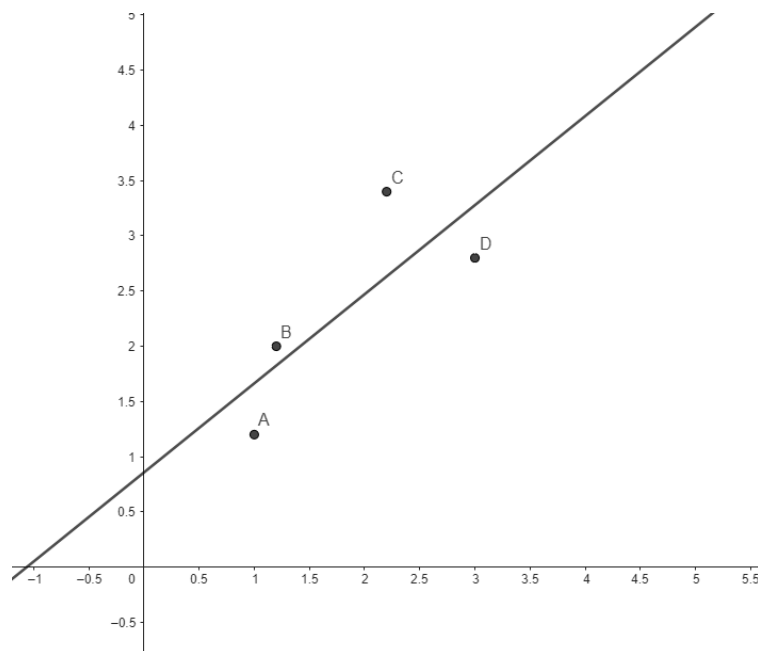


Abbildung 3.8.: Lineare Regression

Multiple Linear Regression

- Eine abhängige Variable
- Mehrere unabhängige Variablen zur Bestimmung der Abhängigen

(22)

Das klassische Modell der linearen Mehrfachregression wird durch folgende Formel dargestellt:

$$y_t = x_{t1}\beta_1 + x_{t2}\beta_2 + \dots + x_{tK}\beta_K + \varepsilon_t$$

(22)

In Matrixschreibweise lautet das Modell:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

(22)

Variablenbeschreibung

- y_t : Abhängige Variable
- \mathbf{y} : Vektor der abhängigen Variablen ($T \times 1$).
- x_{tk} : Werte der k -ten unabhängigen (exogenen) Variable für die Beobachtung t ($k = 1, 2, \dots, K$).
- \mathbf{X} : Designmatrix der unabhängigen Variablen ($T \times K$), wobei jede Zeile die Werte der unabhängigen Variablen für eine Beobachtung enthält.
- β_k : Regressionsparameter (Koeffizienten), die den Einfluss der k -ten unabhängigen Variable auf die abhängige Variable quantifizieren.
- $\boldsymbol{\beta}$: Vektor der Regressionsparameter ($K \times 1$).
- ε_t : Störgröße (Fehlerterm) für die Beobachtung t , die zufällig ist und die Abweichung des Modells von den tatsächlichen Werten beschreibt.
- $\boldsymbol{\varepsilon}$: Vektor der Störgrößen ($T \times 1$).
- T : Anzahl der Beobachtungen.
- K : Anzahl der unabhängigen Variablen.

(22)

3.3.2. Ansätze und Anwendungen der Regressionsanalyse

Ein vielversprechender Ansatz in der linearen multiplen Regression besteht im Vergleich der standardisierten Koeffizienten. Dabei wird der **Regressionskoeffizient** als Maß für den Einfluss einer unabhängigen Variable auf die abhängige Variable interpretiert. Durch die Standardisierung der einzelnen Koeffizienten lässt sich deren relative Wirkung direkt vergleichen, sodass sich die unabhängige Variable ermitteln lässt, die den größten Einfluss auf das Gesamtergebnis ausübt.(23)

3.4. Anforderungen an eine skalierbare Datenverarbeitungsarchitektur

3.4.1. Anforderungen an den Datenfluss für Finanzdaten

Der Datenfluss in Finanzsystemen muss hohe Anforderungen hinsichtlich Geschwindigkeit, Skalierbarkeit und Datenintegrität erfüllen, um die Verarbeitung großer Datenmengen effizient zu gewährleisten. Finanzmärkte generieren kontinuierlich umfangreiche Transaktionsdaten, historische Marktinformationen und Echtzeit-Kurse, die in Handelssystemen verarbeitet werden müssen. Um diese Datenströme effizient zu verwalten, sind durchdachte ETL-Architekturen (Extrahieren, Transformieren, Laden) erforderlich, die sowohl historische als auch Echtzeitdaten verarbeiten können.(24)

Ein effektiver Datenfluss beginnt mit der **Datenextraktion**, bei der Daten aus verschiedenen Quellen wie Börsen, Banken oder Nachrichtenfeeds aggregiert werden. Hierbei ist es essenziell, dass die Schnittstellen standardisierte Formate verwenden und möglichst geringe Latenzzeiten aufweisen. Anschließend folgt die **Datenverarbeitung**, in der Bereinigung, Validierung und Aggregation erfolgen. Dies dient dazu, fehlerhafte oder unvollständige Daten zu korrigieren und eine einheitliche Struktur für Analysen bereitzustellen. Abschließend werden die transformierten Daten in eine Datenbank geladen, wo sie für algorithmische Handelsmodelle und weiterführende Analysen verfügbar sind.(24)

3.4.2. Integrationsanforderungen für technische und analytische Systeme

Die Integration technischer und analytischer Systeme spielt eine zentrale Rolle in der modernen Datenverarbeitung, insbesondere bei der Analyse und Verarbeitung großer Finanzdatensätze.


Definition technischer und analytischer Systeme

Technische und analytische Systeme unterscheiden sich in ihrer Funktion und ihrem Zweck, müssen jedoch nahtlos zusammenarbeiten, um eine effiziente Datenverarbeitung und -analyse zu gewährleisten.

Technische Systeme umfassen die grundlegenden Infrastrukturen und Prozesse, die für die Speicherung, Verarbeitung und Bereitstellung von Daten erforderlich sind. Dazu gehören:

- Datenbanken wie MySQL oder PostgreSQL zur strukturierten Speicherung großer Datenmengen.
- ETL-Prozesse (Extract, Transform, Load) zur Extraktion, Transformation und Speicherung von Daten aus verschiedenen Quellen.
- APIs zur Bereitstellung von Daten für externe Anwendungen und Systeme.
- Verteilte Rechensysteme wie Apache Spark, die große Datenmengen parallel verarbeiten können und für Big-Data-Anwendungen essenziell sind.

Analytische Systeme hingegen sind darauf ausgelegt, Daten auszuwerten, Muster zu erkennen und fundierte Entscheidungen zu unterstützen. Zu diesen Systemen gehören:

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

- Machine-Learning-Modelle, die Muster in historischen Marktdaten erkennen und Vorhersagen ermöglichen.
- Statistische Analysen, die Beziehungen zwischen verschiedenen Marktindikatoren analysieren.
- Visualisierungswerkzeuge wie .Net Blazor, die komplexe Daten in verständliche Grafiken und Dashboards umwandeln.

(25)

Integrationsanforderungen

Die Integration erfordert die Berücksichtigung spezifischer Anforderungen, um einen nahtlosen Datenfluss und eine effiziente Verarbeitung sicherzustellen. Ein zentrales Element dabei ist die Erstellung eines **Integrationsanforderungsdokuments**, das die Erwartungen und Funktionen des Integrationsprojekts detailliert beschreibt. Dieses Dokument dient als Grundlage für die erfolgreiche Umsetzung der Datenintegration und gewährleistet die Kompatibilität der beteiligten Systeme. (26)

Ein weiterer wichtiger Aspekt ist die Sicherstellung der **Interoperabilität** zwischen den Systemen. Interoperabilität bezeichnet die Fähigkeit verschiedener Systeme, nahtlos zusammenzuarbeiten und Daten effizient auszutauschen, ohne dass besondere Anpassungen erforderlich sind. Dies setzt die Einhaltung gemeinsamer technischer Normen und Standards voraus, um eine reibungslose Kommunikation und Integration zu ermöglichen. (27)

3.5. Funktionsweise und Architektur von Apache Spark

3.5.1. Grundkonzepte und Datenverarbeitung in Spark

Apache Spark ist ein leistungsstarkes Open-Source-Framework für die verteilte Verarbeitung großer Datenmengen. Es wurde entwickelt, um die Einschränkungen des traditionellen MapReduce-Modells zu überwinden und bietet sowohl In-Memory- als auch Festplattenbasierte Datenverarbeitung. Die Hauptkomponenten von Spark umfassen Spark Core, Spark SQL, Spark Streaming, MLlib und GraphX, die zusammen eine flexible und skalierbare Plattform für verschiedene Datenverarbeitungsanforderungen bieten. (28)

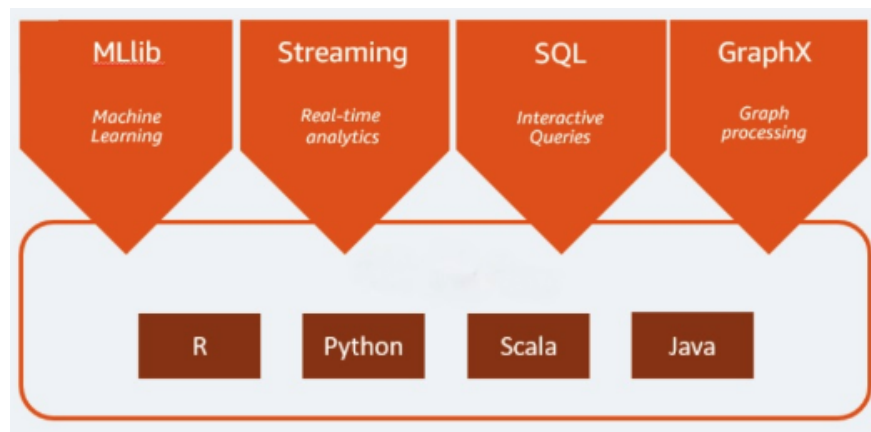



Abbildung 3.9.: Apache Spark

Komponenten von Apache Spark: (28)

- **Spark Core:** Das Herzstück von Spark, das grundlegende Funktionen wie Aufgabenverteilung, Speicherverwaltung und Fehlerbehandlung bereitstellt.
- **Spark SQL:** Ermöglicht die Verarbeitung strukturierter Daten und die Ausführung von SQL-Abfragen auf Spark-Datasets.
- **Spark Streaming:** Unterstützt die Echtzeit-Verarbeitung von Datenströmen, wodurch Anwendungen Daten in nahezu Echtzeit verarbeiten können.
- **MLlib:** Eine Bibliothek für maschinelles Lernen, die gängige Algorithmen und Dienstprogramme für skalierbares maschinelles Lernen bereitstellt.
- **GraphX:** Bietet Werkzeuge für die Verarbeitung und Analyse von Graphstrukturen, wie sie beispielsweise in sozialen Netzwerken vorkommen.

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

Verteiltes Rechnen und RDDs (28)

Ein zentrales Konzept in Spark ist das des Resilient Distributed Datasets (RDD). RDDs sind unveränderliche, verteilte Sammlungen von Objekten, die über ein Cluster hinweg gespeichert werden. Zu den Vorteilen zählen:

- **Robuste Wiederherstellung:** RDDs gewährleisten eine hohe Ausfallsicherheit, da sie die Abfolge ihrer Transformationen speichern. Fällt ein Knoten im Cluster aus, kann das RDD aus den ursprünglichen Datenquellen neu erstellt werden.
- **Unveränderlichkeit:** Einmal erzeugte RDDs bleiben unverändert. Anstatt bestehende Daten zu modifizieren, werden bei jeder Transformation neue RDDs generiert. Dies stellt sicher, dass frühere Versionen der Daten verfügbar bleiben und keine unbeabsichtigten Änderungen auftreten.

In-Memory-Verarbeitung

Einer der Hauptvorteile von Spark gegenüber traditionellen Big-Data-Frameworks ist seine Fähigkeit zur In-Memory-Verarbeitung. Durch das Halten von Daten im Arbeitsspeicher anstatt auf Festplatten kann Spark die Latenzzeiten erheblich reduzieren und die Verarbeitungsgeschwindigkeit steigern. Dies ist besonders vorteilhaft für Anwendungen, die wiederholten Zugriff auf dieselben Daten erfordern, wie beispielsweise maschinelles Lernen und iterative Datenanalysen. (29)

APIs und Sprachen

Spark bietet APIs für verschiedene Programmiersprachen, darunter Scala, Java, Python und R. Diese Vielseitigkeit ermöglicht es Entwicklern, Spark in ihre bestehenden Workflows zu integrieren und die Sprache zu wählen, die am besten zu ihren Anforderungen passt. Die APIs sind darauf ausgelegt, sowohl Benutzerfreundlichkeit als auch Leistung zu bieten, wodurch Spark für eine breite Palette von Anwendungsfällen geeignet ist. (30)

3.5.2. Eignung und Vorteile für die Finanzdatenverarbeitung

Die Verarbeitung großer Finanzdatensätze stellt hohe Anforderungen an Effizienz, Skalierbarkeit und Echtzeitfähigkeit. Apache Spark bietet als In-Memory-Framework zahlreiche Vorteile für die Finanzbranche, insbesondere bei der Extraktion, Transformation und Analyse historischer und aktueller Marktdaten. Die Nutzung verteilter Verarbeitung und leistungstarker ETL-Mechanismen macht Spark zu einer idealen Wahl für die Finanzdatenverarbeitung. (31)

Hohe Verarbeitungsgeschwindigkeit durch In-Memory Computing

Einer der größten Vorteile von Apache Spark ist die In-Memory-Verarbeitung, die es ermöglicht, Daten direkt im Arbeitsspeicher zu speichern und zu verarbeiten. Dies führt zu erheblich schnelleren ETL-Prozessen im Vergleich zu herkömmlichen diskbasierten Lösungen wie Apache Hadoop oder klassischen relationalen Datenbanken. So konnte beispielsweise Facebook durch den Einsatz von Spark ETL-Prozesse um den Faktor 6 beschleunigen. (31)

Unterstützung unterschiedlicher Datenquellen und Formate

Spark bietet mit dem DataSource API eine breite Unterstützung für verschiedene Datenquellen und Dateiformate. Im Finanzsektor ist dies besonders relevant, da Daten aus unterschiedlichen Quellen wie Börsendatenbanken, APIs (z. B. Yahoo Finance), Data Lakes oder strukturierten CSV- und Parquet-Dateien stammen können. Durch die flexible Anbindung und Unterstützung gängiger Formate lässt sich Spark nahtlos in bestehende Finanzsysteme integrieren. (31)

Echtzeit-Verarbeitung und Streaming-Fähigkeit

Moderne Handelsstrategien erfordern oft die Verarbeitung von Marktdaten in Echtzeit. Apache Spark bietet mit Spark Streaming eine leistungsfähige Lösung zur Echtzeit-Datenverarbeitung. So können Finanzinstitute und algorithmische Trading-Systeme Marktdaten nahezu in Echtzeit analysieren und auf Marktveränderungen schnell reagieren. (31)

Skalierbarkeit und Parallelverarbeitung für große Datenmengen

Finanzmärkte generieren täglich enorme Datenmengen. Spark kann diese durch seine verteilte Verarbeitung effizient bewältigen und horizontal skalieren, indem es Rechenprozesse auf mehrere Cluster-Knoten verteilt. Dies ist entscheidend für die Analyse historischer Marktdaten, die oft Jahrzehnte umfassen und in Petabyte-Größe vorliegen können. (31)

3.5.3. Vergleich von Apache Spark mit alternativen Big-Data-Technologien für die Finanzdatenverarbeitung

Zwei zentrale Verarbeitungsmodelle sind das Batch-Processing und das In-Memory-Processing.

Batch-Processing

Batch-Processing bezieht sich auf die Verarbeitung großer Datenmengen in festgelegten Intervallen. Daten werden gesammelt und in Stapeln (Batches) verarbeitet, oft zu Zeiten geringer Systemauslastung. Dieses Modell eignet sich besonders für Anwendungen, bei denen Echtzeitverarbeitung nicht erforderlich ist, wie z.B. monatliche Berichte oder Lohnabrechnungen. Ein Vorteil des Batch-Processings ist die effiziente Nutzung von Systemressourcen, da große Datenmengen auf einmal verarbeitet werden können. Allerdings kann die Latenzzeit hoch sein, da Ergebnisse erst nach Abschluss des gesamten Batches verfügbar sind. (32)

Vergleich der Verarbeitungsmodelle



 Spark vs Hadoop MapReduce		
Factors	Spark	Hadoop MapReduce
Speed	100x times than MapReduce	Faster than traditional system
Written In	Scala	Java
Data Processing	Batch / real-time / iterative / interactive / graph	Batch processing
Ease of Use	Compact & easier than Hadoop	Complex & lengthy
Caching	Caches the data in-memory & enhances the system performance	Doesn't support caching of data

Abbildung 3.10.: Apache Spark vs. Hadoop

Ein prominentes Beispiel für Batch-Processing ist Apache Hadoop, das Daten in großen Blöcken verarbeitet und nach jeder Verarbeitungsstufe auf die Festplatte schreibt. Dies führt zu höheren Latenzzeiten, insbesondere bei iterativen Prozessen. Im Gegensatz dazu nutzt Apache Spark In-Memory-Processing, wodurch es Daten im RAM hält und somit die Verarbeitungsgeschwindigkeit erheblich steigert. Studien zeigen, dass Spark für bestimmte Aufgaben bis zu 100-mal schneller als Hadoop sein kann. (33)

In der Finanzbranche, wo schnelle Entscheidungen auf Basis großer Datenmengen getroffen werden müssen, bietet In-Memory-Processing klare Vorteile. Die Fähigkeit, Daten nahezu in

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

Echtzeit zu analysieren, ermöglicht es Finanzinstituten, schneller auf Marktveränderungen zu reagieren und Wettbewerbsvorteile zu erzielen.

4. Datenverarbeitungskonzepte in Apache Spark

4.1. Implementierung der technischen Indikatoren in Spark

4.1.1. Auswahl und Implementierung relevanter technischer Indikatoren

Die folgenden Indikatoren wurden in der Programmiersprache Python(34). Die Variablen, welche in den folgenden Funktionen verwendet werden, werden im Konstruktor zugewiesen.

Implementierung des SMA

```

1  def add_sma_columns(self, df):
2      # Sortierung des DataFrames nach dem Datum, um die zeitliche
      ↪ Reihenfolge sicherzustellen.
3      df = df.orderBy(self.date_col)
4
5      # Iteration über jede Fenstergröße in self.windows.
6      for w in self.windows:
7          # Erstellung eines Fensters, das nach self.partition_col
          ↪ partitioniert und innerhalb jeder Gruppe
8          # nach self.date_col sortiert wird; es umfasst die letzten w
          ↪ Zeilen.
9          window_spec = Window.partitionBy(self.partition_col) \
10                           .orderBy(self.date_col) \
11                           .rowsBetween(-(w-1), 0)
12
13          # Erstellung eines neuen Spaltennamen, z.B. "SMA_50" für w = 50.
14          sma_col_name = f"SMA_{w}"
15
16          # Berechnung des Durchschnitts der Werte in self.close_col über das
          ↪ definierte Fenster
17          # und füge diesen als neue Spalte dem DataFrame hinzu.
18          df = df.withColumn(sma_col_name,
          ↪ F.avg(self.close_col).over(window_spec))
19
20          #Rückgabe des Dataframes.
21          return df

```

Listing 4.1.: SMA-Implementierung

Implementierung des EMA

```

1  def add_ema_column(self, df):
2      # Es wird geprüft, ob die Spalten für Schlusskurs, Datum und Partition
      ↳ im DataFrame existieren.
3      for col_name in [self.close_col, self.date_col, self.partition_col]:
4          if col_name not in df.columns:
5              raise ValueError(f"Spalte '{col_name}' nicht im DataFrame
              ↳ gefunden")
6
7      # Der Name für die neue EMA-Spalte wird erstellt, z.B. "EMA_20".
8      ema_col_name = f"EMA_{self.span}"
9
10     # Das ursprüngliche Schema des DataFrames wird erfasst und um die
      ↳ EMA-Spalte erweitert.
11     original_fields = df.schema.fields
12     all_fields = original_fields + [StructField(ema_col_name, DoubleType(),
      ↳ True)]
13     out_schema = StructType(all_fields)
14
15     # Eine Funktion zur Berechnung des EMA wird definiert, die auf ein
      ↳ Pandas DataFrame angewendet wird.
16     def compute_ema(pdf: pd.DataFrame) -> pd.DataFrame:
17         # Das Pandas DataFrame wird nach dem Datum sortiert.
18         pdf = pdf.sort_values(by=self.date_col)
19         # Der exponentielle gleitende Durchschnitt (EMA) der Schlusskurse
      ↳ wird berechnet und in der neuen Spalte gespeichert.
20         pdf[ema_col_name] = pdf[self.close_col].ewm(span=self.span,
      ↳ adjust=False).mean()
21         # Die Spaltenreihenfolge wird an das erweiterte Schema angepasst.
22         output_cols = [field.name for field in out_schema.fields]
23         pdf = pdf[output_cols]
24         return pdf
25
26     # Die compute_ema-Funktion wird gruppenweise (nach self.partition_col)
      ↳ mittels applyInPandas auf den DataFrame angewendet.
27     df_with_ema = df.groupby(self.partition_col).applyInPandas(compute_ema,
      ↳ schema=out_schema)
28     return df_with_ema

```

Listing 4.2.: EMA-Implementierung

Implementierung des Momentum Indikators

```

1  def calculate(self, df):
2      # Erstellung eines Fensters, das nach "TICKER_SYMBOL" partitioniert und
      ↪ nach "DATE" sortiert.
3      window_spec = Window.partitionBy("TICKER_SYMBOL").orderBy("DATE")
4
5      # Berechnung des Momentums, indem vom aktuellen Schlusskurs der
      ↪ Schlusskurs abgezogen wird,
6      # der um self.momentum_period Zeilen zurückliegt; das Ergebnis wird in
      ↪ der Spalte "MOMENTUM" gespeichert.
7      df = df.withColumn("MOMENTUM", col(self.close_col) -
      ↪ F.lag(self.close_col, self.momentum_period).over(window_spec))
8
9      #Rückgabe des neuen Dataframes.
10     return df

```

Listing 4.3.: Momentum-Implementierung

Implementierung des RSI Indikators

```

1  def calculate(self, df):
2      # Definition eines Fensters, das die Daten pro Ticker gruppiert und
      ↪ nach Datum sortiert.
3      window_spec = Window.partitionBy("TICKER_SYMBOL").orderBy("DATE")
4
5      # Berechnung der Kursänderung: Differenz zwischen aktuellem Schlusskurs
      ↪ und dem vorherigen Schlusskurs.
6      df = df.withColumn("RSI(Change)", col(self.close_col) -
      ↪ F.lag(self.close_col).over(window_spec))
7
8      # Erzeugung einer Spalte "RSI(Gain)", die positive Kursänderungen
      ↪ übernimmt, sonst 0.
9      df = df.withColumn("RSI(Gain)", F.when(col("RSI(Change)") > 0,
      ↪ col("RSI(Change)")).otherwise(0))
10
11     # Erzeugung einer Spalte "RSI(Loss)", die negative Kursänderungen
      ↪ übernimmt, sonst 0.
12     df = df.withColumn("RSI(Loss)", F.when(col("RSI(Change)") < 0,
      ↪ col("RSI(Change)")).otherwise(0))
13
14     # Berechnung des gleitenden Durchschnitts der Gewinne über das
      ↪ definierte Fenster.
15     df = df.withColumn("RSI(Avg_Gain)",
      ↪ F.avg("RSI(Gain)").over(window_spec.rowsBetween(-self.period + 1,
      ↪ 0)))
16
17     # Berechnung des gleitenden Durchschnitts der Verluste über das
      ↪ definierte Fenster.
18     df = df.withColumn("RSI(Avg_Loss)",
      ↪ F.avg("RSI(Loss)").over(window_spec.rowsBetween(-self.period + 1,
      ↪ 0)))
19
20     # Errechnung des Verhältnisses (RS) von durchschnittlichen Gewinnen zu
      ↪ durchschnittlichen Verlusten.
21     df = df.withColumn("RSI(RS)", col("RSI(Avg_Gain)") /
      ↪ col("RSI(Avg_Loss)"))
22
23     # Berechnung des RSI-Werts mit der Formel: RSI = 100 - (100 / (1 +
      ↪ RS)).
24     df = df.withColumn("RSI(RSI)", 100 - (100 / (1 + col("RSI(RS)"))))
25
26     return df

```

Listing 4.4.: RSI-Implementierung

Implementierung der Bollinger Bänder

```

1  def calculate(self, df):
2      # Erstellung eines Fensters, das Daten pro Ticker gruppiert und
      ↳ innerhalb jeder Gruppe nach Datum sortiert.
3      window_spec = Window.partitionBy("TICKER_SYMBOL").orderBy("DATE")
4
5      # Berechnung des einfachen gleitenden Durchschnitts (SMA) der
      ↳ Schlusskurse über die letzten self.period Zeilen.
6      df = df.withColumn("Bol(SMA)", F.avg(self.close_col)
7                      .over(window_spec.rowsBetween(-self.period + 1, 0)))
8
9      # Berechnung der Standardabweichungen der Schlusskurse über dasselbe
      ↳ Fenster.
10     df = df.withColumn("Bol(Std_Dev)", F.stddev(self.close_col)
11                     .over(window_spec.rowsBetween(-self.period + 1, 0)))
12
13     # Berechnung des oberen Bollinger Bands: SMA plus 2 mal die
      ↳ Standardabweichung.
14     df = df.withColumn("Bol(Bollinger_Upper)",
15                     col("Bol(SMA)") + 2 * col("Bol(Std_Dev)"))
16
17     # Berechnung des unteren Bollinger Bands: SMA minus 2 mal die
      ↳ Standardabweichung.
18     df = df.withColumn("Bol(Bollinger_Lower)",
19                     col("Bol(SMA)") - 2 * col("Bol(Std_Dev)"))
20
21     # Rückgabe des Dataframes
22     return df

```

Listing 4.5.: Bollinger Bänder-Implementierung

```

1  def calculate(self, df):
2      # Erstellung eines Fensters, das pro "TICKER_SYMBOL" gruppiert und die
      ↪ Daten nach "DATE" sortiert.
3      window_spec = Window.partitionBy("TICKER_SYMBOL").orderBy("DATE")
4
5      # Berechnung des höchsten Kurses der letzten 'period' Zeilen und
      ↪ Speicherung in "Sto(Highest_High)"
6      df = df.withColumn("Sto(Highest_High)",
7
8                          ↪ F.max(self.high_col).over(window_spec.rowsBetween(-self.period,
8                          ↪ + 1, 0)))
9
10     # Berechnung des niedrigsten Kurses der letzten 'period' Zeilen und
      ↪ Speicherung in "Sto(Lowest_Low)"
11     df = df.withColumn("Sto(Lowest_Low)",
12
13                       ↪ F.min(self.low_col).over(window_spec.rowsBetween(-self.period,
13                       ↪ + 1, 0)))
14
15     # Berechnung den %K-Wert:
16     # (%K) = ((aktueller Schlusskurs - Sto(Lowest_Low)) /
      ↪ (Sto(Highest_High) - Sto(Lowest_Low))) * 100
17     df = df.withColumn("Sto(%K)",
18
19                       ↪ (col(self.close_col) - col("Sto(Lowest_Low))) /
19                       ↪ (col("Sto(Highest_High)" - col("Sto(Lowest_Low)))
19                       ↪ * 100)
20
21     # Berechnung den %D-Wert als Durchschnitt von %K über die letzten
      ↪ 'smoothing' Zeilen.
22     df = df.withColumn("Sto(%D)",
23
24                       ↪ F.avg("Sto(%K)").over(window_spec.rowsBetween(-self.smoothing,
24                       ↪ + 1, 0)))
25
26     return df

```

Listing 4.6.: Stochastik-Implementierung

4.2. Implementierung der analytischen Methodiken

4.2.1. Implementierung des Regressionsmodells

Folgender Code zeigt die Implementierung einer multiplen linearen Regression. Als Ergebnis wird der ausschlaggebendste Indikator mithilfe des Koeffizientenvergleichs ermittelt. Diese

Art der Regression eignet sich gut, da sie mehrere unabhängige Variablen (Indikatoren) und eine Abhängige Variable (Profit) berücksichtigt.

```

1 trades_df = pd.DataFrame(all_trade_results)
2
3 trades_df = trades_df.dropna(subset=["profit", "MOMENTUM",
4 "RSI", "EMA_20", "SMA_50", "Bol(Bollinger_Lower)",
5 "Bol(Bollinger_Upper)", "Sto(%K)"])
6
7 feature_columns = ["MOMENTUM", "RSI", "EMA_20", "SMA_50",
8 "Bol(Bollinger_Lower)", "Bol(Bollinger_Upper)", "Sto(%K)"]
9
10 X = trades_df[feature_columns]
11 y = trades_df["profit"]
12
13 scaler = StandardScaler()
14 X_scaled = scaler.fit_transform(X)
15
16 model = LinearRegression()
17 model.fit(X_scaled, y)
18
19 coefficients = model.coef_
20 importance = dict(zip(feature_columns, coefficients))
21
22 most_important = max(importance.items(), key=lambda x: abs(x[1]))

```

Listing 4.7.: Regressionsmodell-Implementierung

Erklärung des Codes:

Zunächst werden die gesammelten Trade-Ergebnisse in ein Pandas DataFrame konvertiert und Zeilen, in denen in den relevanten Spalten (Profit sowie die technischen Indikatorwerte) fehlende Werte vorhanden sind, werden entfernt. Darauffolgend werden die Indikatoren als Features (MOMENTUM, RSI, EMA_20, SMA_50, Bollinger-Bänder und Sto(%K)) bestimmt und der Tradeprofit als abhängige Variable (Target) definiert.

Die Features können unterschiedliche Skalen aufweisen, weshalb sie mit dem StandardScaler standardisiert werden. Dadurch haben alle Variablen einen Mittelwert von 0 und eine Standardabweichung von 1. Anschließend wird ein multiples lineares Regressionsmodell trainiert, um die Beziehung zwischen den standardisierten Features und dem Profit zu modellieren.

Die daraus ermittelten Koeffizienten zeigen an, in welchem Maße sich der Gewinn verändert, wenn der jeweilige Indikator um eine Standardabweichung variiert. Schließlich wird der Indikator, der den höchsten absoluten Koeffizienten aufweist, als ausschlaggebendster Indikator gewertet.

4.3. Implementierung der Datenverarbeitung für historische Marktdaten

4.3.1. Erstmalige Speicherung und Transformation von Finanzmarktdaten in Apache Spark

Für die Analyse historischer Marktdaten wird Apache Spark als verteiltes Datenverarbeitungssystem eingesetzt. Die Verarbeitung wurde in der Programmiersprache Python entwickelt.

Erstellung der Spark Session

```

1  # Erstellung der Spark Session zur verteilten Verarbeitung der Daten
2  spark = SparkSession.builder \
3      .appName("YahooFinanceStockDataLoad") \  # Setzt den Namen der
        ↪ Spark-Anwendung
4      .config("spark.sql.shuffle.partitions", "8") \  # Legt die Anzahl der
        ↪ Shuffle-Partitionen für SQL-Abfragen fest
5      .config("spark.ui.port", "4040") \  # Definiert den UI-Port für das
        ↪ Spark Monitoring
6      .master("local[*]") \  # Nutzt alle verfügbaren CPU-Kerne für die
        ↪ Berechnungen
7      .getOrCreate()  # Erstellt oder holt eine bestehende Spark-Session

```

Listing 4.8.: Erstellung der Spark Session

Abrufen historischer Finanzdaten mit yfinance

```

1  class GetHistoricalDataCommand(Command):
2      def __init__(self, ticker_symbol, start_date, end_date, interval):
3          self.ticker_symbol = ticker_symbol  # Speichert das Ticker-Symbol
4          ↪ des Wertpapiers
5          self.start_date = start_date  # Startdatum für die historischen
6          ↪ Daten
7          self.end_date = end_date  # Enddatum für die historischen Daten
8          self.interval = interval  # Intervall für die Daten (z.B. '1d',
9          ↪ '1h')
10         self.historical_data = None  # Variable zur Speicherung der
11         ↪ abgerufenen Daten
12
13     def execute(self):
14         self._fetch_historical_data()  # Ruft die historischen Daten ab
15         return self.historical_data  # Gibt die abgerufenen Daten zurück
16
17     def _fetch_historical_data(self):
18         for attempt in range(5):  # Fünf Versuche, um Daten abzurufen,
19         ↪ falls Fehler auftreten
20             try:
21                 ticker = yf.Ticker(self.ticker_symbol)  # Erstellt ein
22                 ↪ Ticker-Objekt mit `yfinance`
23                 self.historical_data = ticker.history(
24                     start=self.start_date, end=self.end_date,
25                     ↪ interval=self.interval)  # Holt die historischen
26                     ↪ Kursdaten
27
28                 if not self.historical_data.empty:
29                     return  # Falls Daten vorhanden sind, beendet die
30                     ↪ Methode die Schleife
31             except Exception as e:
32                 time.sleep(10)  # Falls ein Fehler auftritt, wird kurz
33                 ↪ gewartet, bevor erneut versucht wird

```

Listing 4.9.: Abrufen historischer Finanzdaten

Konvertierung der Daten in ein Spark DataFrame

```

1  class SaveToSparkDataFrameCommand(Command):
2      def __init__(self, ticker_symbol, historical_data):
3          self.ticker_symbol = ticker_symbol  # Speichert das Ticker-Symbol
4          self.historical_data = historical_data  # Speichert die
           ↳ historischen Daten als Pandas-DataFrame
5          self.spark_df = None  # Platzhalter für das Spark DataFrame
6
7      def execute(self):
8          self._convert_to_spark_dataframe()  # Konvertiert die Daten in ein
           ↳ Spark DataFrame
9          self._clean_data()  # Bereinigt das DataFrame
10         return self.spark_df  # Gibt das bereinigte DataFrame zurück
11
12     def _convert_to_spark_dataframe(self):
13         # Definiert das Schema für das Spark DataFrame
14         schema = StructType([
15             StructField("date", TimestampType(), True),  # Spalte für das
           ↳ Datum
16             StructField("Open", DoubleType(), True),  # Spalte für den
           ↳ Eröffnungskurs
17             StructField("High", DoubleType(), True),  # Spalte für das
           ↳ Tageshoch
18             StructField("Low", DoubleType(), True),  # Spalte für das
           ↳ Tagestief
19             StructField("Close", DoubleType(), True),  # Spalte für den
           ↳ Schlusskurs
20             StructField("Volume", DoubleType(), True)  # Spalte für das
           ↳ Handelsvolumen
21         ])
22
23         # Konvertiert Pandas DataFrame in Spark DataFrame mit definiertem
           ↳ Schema
24         rows = [
25             (index.to_pydatetime(), float(row.Open), float(row.High),
26              float(row.Low), float(row.Close), float(row.Volume))
27             for index, row in self.historical_data.iterrows()
28         ]
29         self.spark_df = spark.createDataFrame(rows,
           ↳ schema=schema).repartition(8)  # Partitionierung für parallele
           ↳ Verarbeitung

```

```

1      def _clean_data(self):
2          def validate_and_clean_partition(partition):
3              cleaned_rows = []
4              for row in partition:
5                  if all(v is not None for v in row): # Prüft, ob keine
6                  ↪ Nullwerte vorhanden sind
7                      try:
8                          cleaned_row = Row(
9                              date=row[0], Open=float(row[1]),
10                             ↪ High=float(row[2]),
11                             Low=float(row[3]), Close=float(row[4]),
12                             ↪ Volume=float(row[5])
13                          )
14                          cleaned_rows.append(cleaned_row) # Fügt bereinigte
15                          ↪ Zeile zur Liste hinzu
16                      except ValueError as e:
17                          logging.warning(f"Zeile übersprungen aufgrund eines
18                          ↪ Konvertierungsfehlers: {e}") # Loggt Fehler
19              return cleaned_rows
20
21      cleaned_rdd =
22      ↪ self.spark_df.rdd.mapPartitions(validate_and_clean_partition)
23      ↪ # Bereinigt das RDD
24      self.spark_df = spark.createDataFrame(cleaned_rdd,
25      ↪ schema=self.spark_df.schema) # Konvertiert zurück in ein
26      ↪ DataFrame

```

Listing 4.10.: Konvertierung der Daten in ein Spark DataFrame

```

1  class SaveToMySQLCommand(Command):
2      def __init__(self, security_id, spark_df, interval):
3          self.security_id = security_id  # Speichert die Wertpapier-ID aus
           ↳ der Datenbank
4          self.spark_df = spark_df  # Speichert das Spark DataFrame mit den
           ↳ historischen Daten
5          self.interval = interval  # Speichert das Zeitintervall (z. B.
           ↳ '1d', '1h')
6
7      def execute(self):
8          self._save_to_mysql()  # Speichert die Daten in MySQL
9
10     def _save_to_mysql(self):
11         try:
12             connection = pymysql.connect(  # Verbindung zur MySQL-Datenbank
13                 ↳ aufbauen
14                 host='db',
15                 user='root',
16                 password='123mysql',
17                 database='acatradbotdb'
18             )
19             cursor = connection.cursor()
20
21             data_to_insert = self._prepare_data_for_insertion(cursor)  #
22                 ↳ Daten für das Einfügen vorbereiten
23
24             if data_to_insert:
25                 cursor.executemany(
26                     "INSERT INTO HISTORICAL_PRICES (SECURITY_ID, DATE,
27                     ↳ OPEN, CLOSE, HIGH, LOW, VOLUME, CHART_INTERVAL)
28                     ↳ VALUES (%s, %s, %s, %s, %s, %s, %s, %s)",
29                     data_to_insert
30                 )
31                 connection.commit()  # Änderungen in der Datenbank
32                     ↳ speichern
33
34         finally:
35             cursor.close()  # Cursor schließen
36             connection.close()  # Verbindung schließen

```

```

1     def _prepare_data_for_insertion(self, cursor):
2         data_to_insert = []
3         for row in self.spark_df.collect(): # Daten in einer Liste für
4             ↪ Batch-Einfügen speichern
5             if not self._record_exists(cursor, row.date): # Prüft, ob das
6                 ↪ Datum bereits existiert
7                 data_to_insert.append((
8                     self.security_id, row.date, row.Open, row.Close,
9                     ↪ row.High, row.Low, row.Volume, self.interval
10                ))
11         return data_to_insert
12
13     def _record_exists(self, cursor, date):
14         cursor.execute(
15             "SELECT COUNT(*) FROM HISTORICAL_PRICES WHERE SECURITY_ID = %s
16             ↪ AND DATE = %s AND CHART_INTERVAL = %s",
17             (self.security_id, date, self.interval)
18         )
19         count = cursor.fetchone()[0]
20         return count > 0 # Gibt True zurück, falls der Eintrag bereits
21             ↪ existiert

```

Listing 4.11.: Speicherung der Daten in MySQL


Erklärung der Funktionsweise von dataCreate.py

Der Code in dataCreate.py dient dazu, historische Marktdaten für eine Vielzahl von Wertpapieren aus der Yahoo Finance API abzurufen, zu verarbeiten und in einer MySQL-Datenbank zu speichern. Dieses Skript wird vor allem dann verwendet, wenn noch keine Daten in der Datenbank vorhanden sind. Es holt historische Daten für verschiedene Zeitintervalle (z. B. stündlich, täglich, wöchentlich, monatlich) und speichert sie strukturiert in der Datenbank.

Das Skript nutzt Apache Spark, um die großen Datenmengen effizient zu verarbeiten und zu bereinigen. Die verteilte Verarbeitung ermöglicht es, auch umfangreiche Finanzdatensätze performant zu transformieren. Nach der Verarbeitung werden die bereinigten Daten in MySQL eingefügt, sodass sie für spätere Analysen oder Handelsentscheidungen verwendet werden können.

4.3.2. Kontinuierliche Aktualisierung der Finanzmarktdaten in Apache Spark

Das dataUpdate.py-Skript dient der kontinuierlichen Aktualisierung der Finanzmarktdaten in der MySQL-Datenbank. Es wird regelmäßig ausgeführt, um neue Kursdaten aus Yahoo Finance abzurufen und in die Datenbank einzufügen. Dabei wird zunächst geprüft, welches

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

das letzte gespeicherte Datum für jedes Wertpapier ist, und es werden nur Daten ab diesem Zeitpunkt abgerufen.

Dieses Skript ist effizienter als dataCreate.py, da es nur fehlende oder neue Daten einfügt, anstatt den gesamten Datensatz erneut zu laden. Dies reduziert die Rechenlast erheblich und ermöglicht eine schnelle Aktualisierung, ohne doppelte Einträge zu erzeugen.

Wann wird dataUpdate.py verwendet?

- Regelmäßige Datenaktualisierung: Neue Finanzdaten werden hinzugefügt, ohne die gesamte Datenbank neu aufzubauen.
- Effiziente Verarbeitung: Durch das Abrufen nur der neuesten Werte bleibt die Datenbank aktuell, ohne unnötigen Rechenaufwand.
- Automatisierte Marktanalyse: Ermöglicht Echtzeit- oder periodische Analysen auf aktuellen Daten für den Trading-Bot.

Erstellung der Spark-Session für die verteilte Verarbeitung

```

1  # Erstellung der Spark Session zur verteilten Verarbeitung der
   ↪ Finanzmarktdaten
2  spark = SparkSession.builder \
3      .appName("YahooFinanceStockDataUpdate") \ # Name der Spark-Anwendung
4      .config("spark.sql.shuffle.partitions", "8") \ # Anzahl der
   ↪ Partitionen für die Verteilung von Daten
5      .config("spark.ui.port", "4040") \ # Festlegung eines UI-Ports zur
   ↪ Überwachung der Spark-Jobs
6      .master("local[*]") \ # Nutzung aller verfügbaren CPU-Kerne auf dem
   ↪ lokalen Rechner
7      .getOrCreate()

```

Listing 4.12.: Erstellung der Spark-Session für die verteilte Verarbeitung

Abruf der gespeicherten Wertpapiere aus der MySQL-Datenbank

```

1  class GetSecuritiesCommand(Command):
2      def __init__(self):
3          self.securities = []  # Liste zur Speicherung der abgerufenen
                                ↳ Wertpapiere
4
5      def execute(self):
6          self._get_securities()
7          return self.securities
8
9      def _get_securities(self):
10         try:
11             # Verbindung zur MySQL-Datenbank herstellen
12             connection = pymysql.connect(
13                 host='db',
14                 user='root',
15                 password='123mysql',
16                 database='acatradingbotdb',
17                 cursorclass=pymysql.cursors.DictCursor # Abruf der Daten
                                ↳ als Dictionary für einfachere Verarbeitung
18             )
19             with connection.cursor() as cursor:
20                 # SQL-Abfrage: Abrufen aller gespeicherten Wertpapiere mit
                                ↳ ID und Ticker-Symbol
21                 cursor.execute("SELECT SECURITY_ID, TICKER_SYMBOL FROM
                                ↳ SECURITIES")
22                 self.securities = cursor.fetchall() # Speichern der
                                ↳ Ergebnisse in der Liste
23                 logging.info(f"{len(self.securities)} Securities aus der
                                ↳ Datenbank abgerufen.") # Log-Ausgabe der Anzahl der
                                ↳ Wertpapiere
24             except pymysql.MySQLError as err:
25                 logging.error(f"Fehler beim Abrufen der Securities: {err}") #
                                ↳ Fehlerprotokollierung, falls ein Fehler auftritt
26         finally:
27             if connection:
28                 connection.close() # Sicherstellen, dass die
                                ↳ Datenbankverbindung geschlossen wird
29

```

Listing 4.13.: Abruf der gespeicherten Wertpapiere aus der MySQL-Datenbank

```

1  class GetLastDateCommand(Command):
2      def __init__(self, security_id, interval):
3          self.security_id = security_id  # ID des Wertpapiers
4          self.interval = interval  # Zeitintervall (1h, 1d, 1wk, 1mo)
5          self.last_date = None  # Variable zur Speicherung des letzten
           ↳ Datums
6
7      def execute(self):
8          self._get_last_date()
9          return self.last_date
10
11     def _get_last_date(self):
12         try:
13             # Verbindung zur MySQL-Datenbank herstellen
14             connection = pymysql.connect(
15                 host='db',
16                 user='root',
17                 password='123mysql',
18                 database='acatradingbotdb'
19             )
20             with connection.cursor() as cursor:
21                 # Abfrage des zuletzt gespeicherten Datums für das
22                 ↳ Wertpapier und Zeitintervall
23                 cursor.execute("""
24                     SELECT MAX(date) FROM HISTORICAL_PRICES
25                     WHERE SECURITY_ID = %s AND CHART_INTERVAL = %s
26                 """, (self.security_id, self.interval))
27                 result = cursor.fetchone()
28                 self.last_date = result[0] if result[0] else None  #
29                 ↳ Speichern des letzten gespeicherten Datums
30             logging.info(f"Letztes Datum für Security ID {self.security_id}
31                 ↳ und Intervall {self.interval}: {self.last_date}")
32         except pymysql.MySQLError as err:
33             logging.error(f"Fehler beim Abrufen des letzten Datums: {err}")
34             ↳ # Fehlerbehandlung
35         finally:
36             if connection:
37                 connection.close()  # Verbindung schließen

```

Listing 4.14.: Abruf des zuletzt gespeicherten Kursdatums für ein Wertpapier

```

1  class GetNewHistoricalDataCommand(Command):
2      def __init__(self, ticker_symbol, start_date, end_date, interval):
3          self.ticker_symbol = ticker_symbol  # Aktien-Ticker (z.B. AAPL,
4          ↪      TSLA)
5          self.start_date = start_date  # Startdatum für den Abruf
6          self.end_date = end_date  # Enddatum für den Abruf
7          self.interval = interval  # Zeitintervall der Daten (1h, 1d, etc.)
8          self.historical_data = pd.DataFrame()  # DataFrame zur Speicherung
9          ↪      der abgerufenen Daten
10
11     def execute(self):
12         self._fetch_historical_data()
13         return self.historical_data
14
15     def _fetch_historical_data(self):
16         logging.info(f"Abrufen neuer Daten für {self.ticker_symbol} von
17         ↪      {self.start_date} bis {self.end_date} mit Intervall
18         ↪      {self.interval}")
19         ticker = yf.Ticker(self.ticker_symbol)
20         try:
21             # Abruf der historischen Daten aus Yahoo Finance API
22             self.historical_data = ticker.history(start=self.start_date,
23             ↪      end=self.end_date, interval=self.interval)
24             logging.info(f"{len(self.historical_data)} neue Datenreihen für
25             ↪      {self.ticker_symbol} abgerufen.")
26         except Exception as e:
27             logging.error(f"Fehler beim Abrufen der Daten für
28             ↪      {self.ticker_symbol}: {e}")  # Fehlerprotokollierung
29

```

Listing 4.15.: Abruf neuer Finanzmarktdaten über Yahoo Finance

```

1  class ProcessDataWithSparkCommand(Command):
2      def __init__(self, ticker_symbol, historical_data, spark):
3          self.ticker_symbol = ticker_symbol  # Symbol der Aktie
4          self.historical_data = historical_data  # Neue Finanzdaten
5          self.spark = spark  # Spark Session für verteilte Verarbeitung
6          self.spark_df = None  # Spark DataFrame zur Speicherung der
           ↪ verarbeiteten Daten
7
8      def execute(self):
9          self._convert_to_spark_dataframe()
10         self._clean_data_parallel()
11         return self.spark_df
12
13     def _convert_to_spark_dataframe(self):
14         logging.info(f"Konvertiere neue Daten für {self.ticker_symbol} in
           ↪ ein Spark DataFrame...")
15         schema = StructType([
16             StructField("date", TimestampType(), True),
17             StructField("Open", DoubleType(), True),
18             StructField("High", DoubleType(), True),
19             StructField("Low", DoubleType(), True),
20             StructField("Close", DoubleType(), True),
21             StructField("Volume", DoubleType(), True)
22         ])
23         rows = [
24             (index.to_pydatetime(), float(row.Open), float(row.High),
           ↪ float(row.Low), float(row.Close), float(row.Volume))
25             for index, row in self.historical_data.iterrows()
26         ]
27         self.spark_df = self.spark.createDataFrame(rows,
           ↪ schema=schema).repartition(8)  # Partitionierung für parallele
           ↪ Verarbeitung
28

```

Listing 4.16.: Umwandlung der neuen Finanzmarktdaten in ein Spark DataFrame

5. Bewertung der Untersuchungsergebnisse

5.1. Bewertung der Spark Verarbeitungsgeschwindigkeit

5.1.1. Testaufbau und Rahmenbedingungen

In dieser Testreihe wurde die Performance von Apache Spark innerhalb einer Docker-Umgebung untersucht. Dabei liefen sämtliche Tests im local-Modus (Spark master("local[*]")) bei unterschiedlich großen Wertpapiermengen (30, 60 und 120). Die **Zeitmessung** erfolgte über `time.time()` im Python-Skript, während Docker stats zur Überwachung von **CPU- und Speicherverbrauch** diente. Zusätzlich wurde die **Spark UI** (Executors-, Jobs-Ansicht) für Detailanalysen zu Tasks, Shuffle-Operationen und Partitionierung herangezogen.

Messung der Verarbeitungszeit mittels `time.time()`

```

1  def main():
2      start_time = time.time() # Zeitpunkt am Anfang
3      # Liste der zu verarbeitenden Aktien
4      # stocks = [
5      #     {"ticker_symbol": "AAPL", "security_name": "Apple Inc.",
6      #       ↪ "security_type": "Share", "currency": "Euro"},
7      #     {"ticker_symbol": "MSFT", "security_name": "Microsoft Corporation",
8      #       ↪ "security_type": "Share", "currency": "Euro"},
9      #     {"ticker_symbol": "GOOGL", "security_name": "Alphabet Inc.",
10      #      ↪ "security_type": "Share", "currency": "Euro"}, ... ];
11
12     for stock_info in stocks:
13         processor = StockDataProcessor(stock_info)
14         processor.process(start_dates, end_date)
15
16     spark.stop()
17     end_time = time.time() # Zeitpunkt am Ende
18     total_duration = end_time - start_time # Zeitdauer Anfang - Ende
19     logging.info(f"Gesamte Laufzeit aller Aktien: {total_duration:.2f}
20     ↪ Sekunden")
21     # Gibt die Sekunden der Gesamten Laufzeit aus
22
23 if __name__ == "__main__":
24     main()
25

```

Listing 5.1.: Messung der Verarbeitungszeit mittels `time.time()`

5.1.2. Messmethodik

Für die Bewertung der Spark-Performance wurden mehrere Durchläufe mit einer unterschiedlichen Anzahl an Wertpapieren durchgeführt. Die Messungen umfassen:

- Die gesamte Verarbeitungszeit von Spark (vom Laden der Daten bis zur Speicherung in der Datenbank).
- Die Veränderung der Verarbeitungszeit bei einer Verdopplung bzw. Vervielfachung der Datenmenge.
- Die CPU- und Speicherauslastung während der Verarbeitung.

Die Messungen wurden unter identischen Bedingungen durchgeführt, um eine möglichst objektive Analyse zu ermöglichen.

5.1.3. Ergebnisse und Kennzahlen

30 Wertpapiere

- Gesamtlaufzeit: 110,25 Sekunden
- Gesamte Tasks: 2760
- Shuffle Read/Write: 2,7 MiB
- CPU-Auslastung: 646,32
- Speichernutzung: 2,113 GiB

60 Wertpapiere

- Gesamtlaufzeit: 233,56 Sekunden
- Gesamte Tasks: 5568
- Shuffle Read/Write: 4,8 MiB
- CPU-Auslastung: 743,66
- Speichernutzung: 2,107 GiB

120 Wertpapiere

- Gesamtlaufzeit: 372,98 Sekunden
- Gesamte Tasks: 10.864
- Shuffle Read/Write: 8,4 MiB
- CPU-Auslastung: 781,73
- Speichernutzung: 1,983 GiB

Veränderung der Verarbeitungszeit

Anhand der gesammelten Daten kann eine lineare Steigerung der Verarbeitungszeit beobachtet werden:

Anzahl Wertpapiere	Laufzeit (s)	Faktor der Erhöhung
30	110,25	1,00
60	233,56	2,12
120	372,98	3,38

Tabelle 5.1.: Veränderung der Verarbeitungszeit von Apache Spark

Die Verarbeitungszeit steigt nicht exakt linear mit der Datenmenge, sondern leicht überproportional. Dies könnte auf Overhead in der Task-Zuweisung oder eine zunehmende Belastung der Speicherverwaltung zurückzuführen sein.

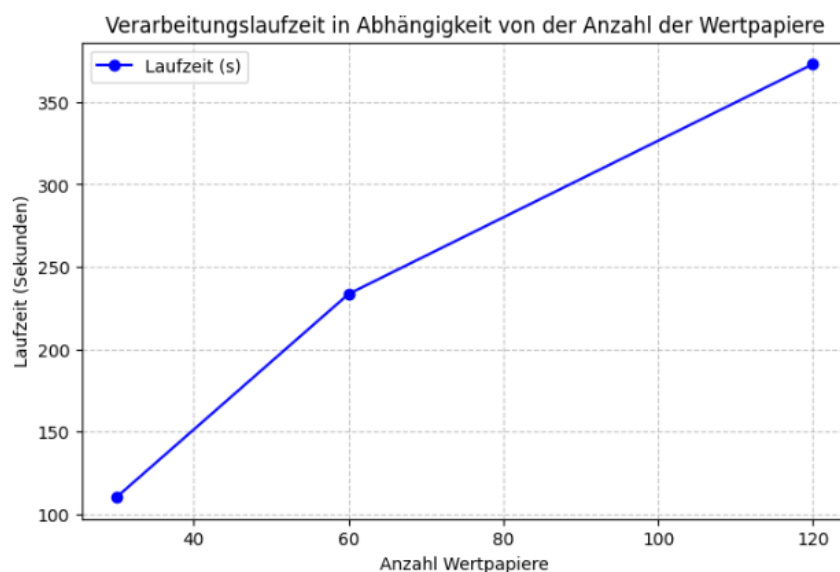


Abbildung 5.1.: Verarbeitungszeit bei steigender Last


CPU- und Speicherauslastung

Die CPU-Auslastung nimmt mit der Anzahl der Wertpapiere zu, zeigt jedoch eine gewisse Stabilität. Während die Speicherbelastung nicht signifikant ansteigt, könnte eine größere Datenmenge möglicherweise zu Speicherproblemen führen, insbesondere wenn der verfügbare RAM begrenzt ist.

Anzahl Wertpapiere	CPU (%)	Speicher (GiB)
30	646,32	2,113
60	743,66	2,107
120	781,73	1,983

Tabelle 5.2.: CPU- und Speicherauslastung bei steigender Anzahl an Wertpapieren

Interessanterweise bleibt die Speichernutzung relativ konstant, obwohl die Anzahl der verarbeiteten Wertpapiere steigt. Dies deutet darauf hin, dass Spark effizient mit Speicherressourcen umgeht und unnötige Daten entfernt oder komprimiert.

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

5.2. Analyse der Implementierten Strategien und deren Aufbau

5.2.1. Vergleich der unterschiedlichen Strategieimplementierungen

Die 3 Strategien wurden so konzipiert, dass Kauf- und Verkaufssignale ausschließlich dann ausgelöst werden, wenn die definierten Regeln der berechneten Indikatoren erfüllt sind. Der Unterschied der Tradingstrategien ergibt sich dabei einerseits aus der **variierenden Gewichtung** der Chartintervalle und andererseits aus der Anpassung der **Restriktivität** der technischen Indikatoren.

Die Werte der Indikatoren sind bereits im Dataframe unter den Spaltennamen vorhanden.

Generelle Strategiestruktur:

```

1  def __init__(self):
2      # Initialisierung: Festlegen der Gewichtungen für verschiedene
      ↳ Zeiträumen
3      self.weights = {
4          "1h": 0.05, # Gewichtung für das 1-Stunden-Intervall
5          "1d": 0.15, # Gewichtung für das 1-Tages-Intervall
6          "1wo": 0.3, # Gewichtung für das 1-Wochen-Intervall
7          "1mo": 0.5 # Gewichtung für das 1-Monats-Intervall
8      }
9      self.buy_signal_threshold = 0.25
10     self.sell_signal_threshold = 0.25
11
12 def interpret_signals(self, df):
13     # --- Kaufentscheidungen für verschiedene Zeiträumen ---
14
15     # Kauf-Signal für 1-Stunden-Intervall (CHART_INTERVAL == 0)
16     df = df.withColumn(
17         "Buy_Signal_1h",
18         when(
19             (col("CHART_INTERVAL") == 0) & # Bedingung: Intervall
20             ↳ entspricht 1h
21             (col("Momentum") > -10) & # Momentum ist oberhalb einer
22             ↳ weniger strengen Schwelle
23             (col("EMA_20") > col("SMA_50")) & # Kurzfristiger EMA (20)
24             ↳ liegt über dem SMA (50)
25             ((col("RSI(RSI)") < 40) | (col("CLOSE") <
26             ↳ col("Bol(Bollinger_Lower)") + 2)), # Entweder RSI unter 40
27             ↳ oder CLOSE nahe dem unteren Bollinger-Band
28             1 # Setze Kauf-Signal
29         ).otherwise(0) # Ansonsten kein Signal
30     )
31
32     # Kauf-Signal für 1-Tages-Intervall (CHART_INTERVAL == 1)
33     df = df.withColumn(
34         "Buy_Signal_1d",
35         when(
36             (col("CHART_INTERVAL") == 1) & # Bedingung: Intervall
37             ↳ entspricht 1d
38             (col("Momentum") > -10) & # Momentum über -10
39             (col("EMA_20") > col("SMA_50")) & # EMA(20) liegt über SMA(50)
40             ((col("RSI(RSI)") < 40) | (col("Sto(%K)") > 70)), # Entweder
41             ↳ RSI unter 40 oder Stochastik(%K) über 70
42             1 # Setze Kauf-Signal
43         ).otherwise(0)
44     )
45
46     # Kauf-Signal für 1-Wochen-Intervall (CHART_INTERVAL == 2)
47     df = df.withColumn(
48         "Buy_Signal_1wo",

```

Parameter der unterschiedlichen Strategien:

Strategie 1

Chartintervalle:

```

1  self.weights = {
2      "1h": 0.05,
3      "1d": 0.15,
4      "1wo": 0.3,
5      "1mo": 0.5
6  }
7  self.buy_signal_threshold = 0.25
8  self.sell_signal_threshold = 0.25

```

Listing 5.3.: Parameter der Strategie 1

Indikatoren Triggerintervalle:

- **Momentum:**
 - Kauf: > -10 (1h, 1d, 1wo)
 - Verkauf: < 10 (1h, 1d, 1wo)
- **EMA_20 vs. SMA_50 (1h, 1d):**
 - Kauf: $\text{EMA}_{20} > \text{SMA}_{50}$
 - Verkauf: $\text{EMA}_{20} < \text{SMA}_{50}$
- **SMA_50 vs. SMA_100 (1wo):**
 - Kauf: $\text{SMA}_{50} > \text{SMA}_{100}$
 - Verkauf: $\text{SMA}_{50} < \text{SMA}_{100}$
- **SMA_50 vs. SMA_200 (1mo):**
 - Kauf: $\text{SMA}_{50} > \text{SMA}_{200}$
 - Verkauf: $\text{SMA}_{50} < \text{SMA}_{200}$
- **RSI:**
 - 1h & 1d: Kauf: < 40 , Verkauf: > 60
 - 1wo & 1mo: Kauf: < 50 , Verkauf: > 55
- **Bollinger Bands (1h):**
 - Kauf: $\text{CLOSE} < \text{Bollinger Lower} + 2$
 - Verkauf: $\text{CLOSE} > \text{Bollinger Upper} - 2$
- **Stochastik (%K) (1d):**
 - Kauf: > 70
 - Verkauf: < 30

Strategie 2

Chartintervalle:

```

1  def __init__(self):
2      self.weights = {
3          "1h": 0.1,
4          "1d": 0.2,
5          "1wo": 0.3,
6          "1mo": 0.4
7      }
8      self.buy_signal_threshold = 0.35
9      self.sell_signal_threshold = 0.35
10

```

Listing 5.4.: Parameter der Strategie 2

Indikatoren Triggerintervalle:

- **Momentum:**
 - Kauf: > 0 (1h, 1d, 1wo)
 - Verkauf: < 0 (1h, 1d, 1wo)
- **EMA_20 vs. SMA_50 (1h, 1d):**
 - Kauf: $\text{EMA}_{20} > \text{SMA}_{50}$
 - Verkauf: $\text{EMA}_{20} < \text{SMA}_{50}$
- **SMA_50 vs. SMA_100 (1wo):**
 - Kauf: $\text{SMA}_{50} > \text{SMA}_{100}$
 - Verkauf: $\text{SMA}_{50} < \text{SMA}_{100}$
- **SMA_50 vs. SMA_200 (1mo):**
 - Kauf: $\text{SMA}_{50} > \text{SMA}_{200}$
 - Verkauf: $\text{SMA}_{50} < \text{SMA}_{200}$
- **RSI:**
 - 1h & 1d: Kauf: < 30 , Verkauf: > 70
 - 1wo & 1mo: Kauf: < 35 , Verkauf: > 65
- **Bollinger Bands (1h):**
 - Kauf: $\text{CLOSE} < \text{Bollinger Lower} + 1$
 - Verkauf: $\text{CLOSE} > \text{Bollinger Upper} - 1$
- **Stochastik (%K) (1d):**
 - Kauf: > 70
 - Verkauf: < 30

Strategie 3

Chartintervalle:

```


1  def __init__(self):
2      self.weights = {
3          "1h": 0.05,
4          "1d": 0.15,
5          "1wo": 0.3,
6          "1mo": 0.5
7      }
8      self.buy_signal_threshold = 0.20
9      self.sell_signal_threshold = 0.20
10

```

Listing 5.5.: Parameter der Strategie 3

Indikatoren Triggerintervalle:

- **Momentum:**
 - Kauf: > -20 (1h, 1d, 1wo)
 - Verkauf: < 20 (1h, 1d, 1wo)
- **EMA_20 vs. SMA_50 (1h, 1d):**
 - Kauf: $\text{EMA}_{20} > \text{SMA}_{50}$
 - Verkauf: $\text{EMA}_{20} < \text{SMA}_{50}$
- **SMA_50 vs. SMA_100 (1wo):**
 - Kauf: $\text{SMA}_{50} > \text{SMA}_{100}$
 - Verkauf: $\text{SMA}_{50} < \text{SMA}_{100}$
- **SMA_50 vs. SMA_200 (1mo):**
 - Kauf: $\text{SMA}_{50} > \text{SMA}_{200}$
 - Verkauf: $\text{SMA}_{50} < \text{SMA}_{200}$
- **RSI:**
 - 1h & 1d: Kauf: < 40 , Verkauf: > 70
 - 1wo & 1mo: Kauf: < 50 , Verkauf: > 55
- **Bollinger Bands (1h):**
 - Kauf: $\text{CLOSE} < \text{Bollinger Lower} + 2$
 - Verkauf: $\text{CLOSE} > \text{Bollinger Upper} - 2$
- **Stochastik (%K) (1d):**
 - Kauf: > 70
 - Verkauf: < 30

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

5.2.2. Optimierungsmöglichkeiten der Strategien für große Datenmengen

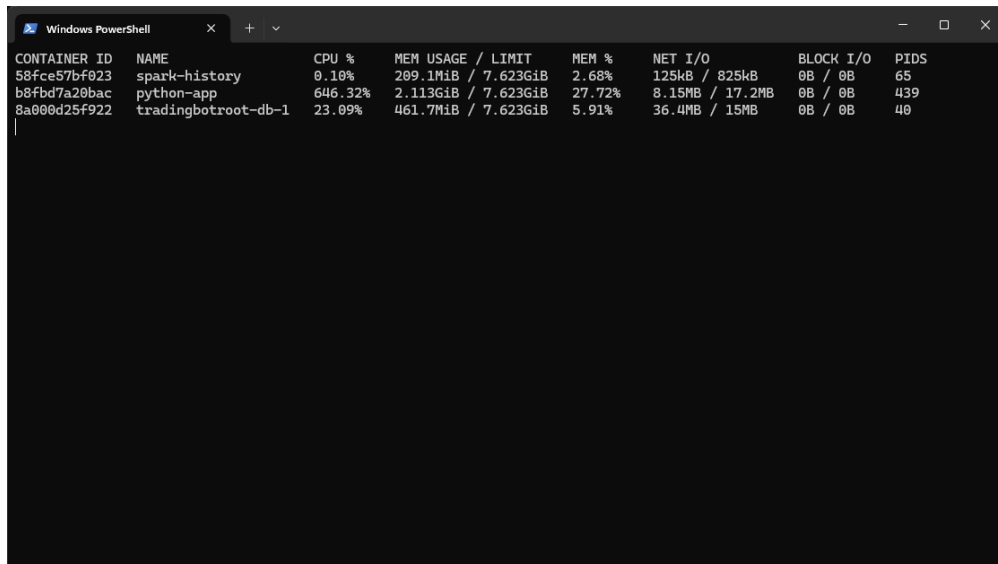
6. Leistungsevaluation und Benchmarking

6.1. Analyse der Verarbeitungsgeschwindigkeit von Apache Spark

6.1.1. Ergebnisse der Messungen: Verarbeitungsgeschwindigkeit von Apache Spark bei historischen Marktdaten

Im Folgenden werden die Ergebnisse der Messungen bei unterschiedlichen Datenmengen dargestellt. Zur Veranschaulichung sind Screenshots aus der Spark UI und den Docker-Statistiken abgebildet, welche die CPU- und Speicherauslastung sowie die verarbeiteten Tasks und Laufzeiten aufzeigen.

30 Wertpapiere:



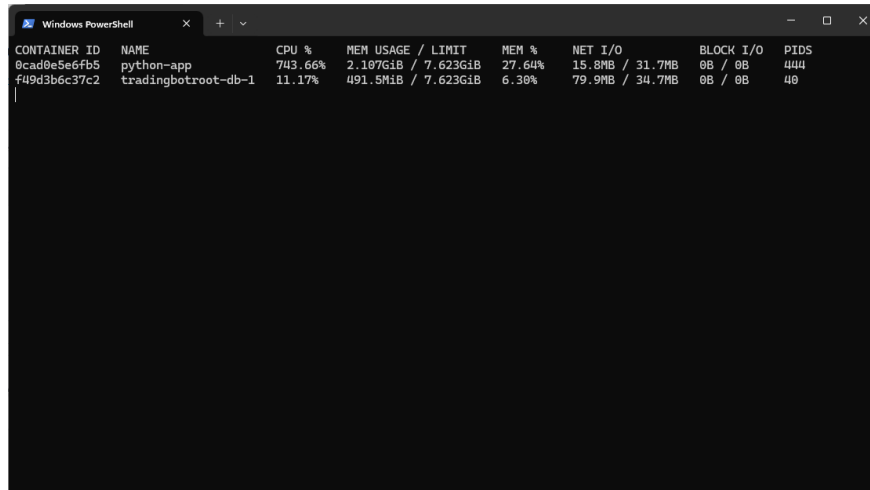
CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
58fce57bf023	spark-history	0.10%	209.1MiB / 7.623GiB	2.68%	125kB / 825kB	0B / 0B	65
b8fbd7a20bac	python-app	646.32%	2.113GiB / 7.623GiB	27.72%	8.15MB / 17.2MB	0B / 0B	439
8a000d25f922	tradingbotroot-db-1	23.09%	461.7MiB / 7.623GiB	5.91%	36.4MB / 15MB	0B / 0B	40

Abbildung 6.1.: Docker Stats bei 30 Wertpapieren

<

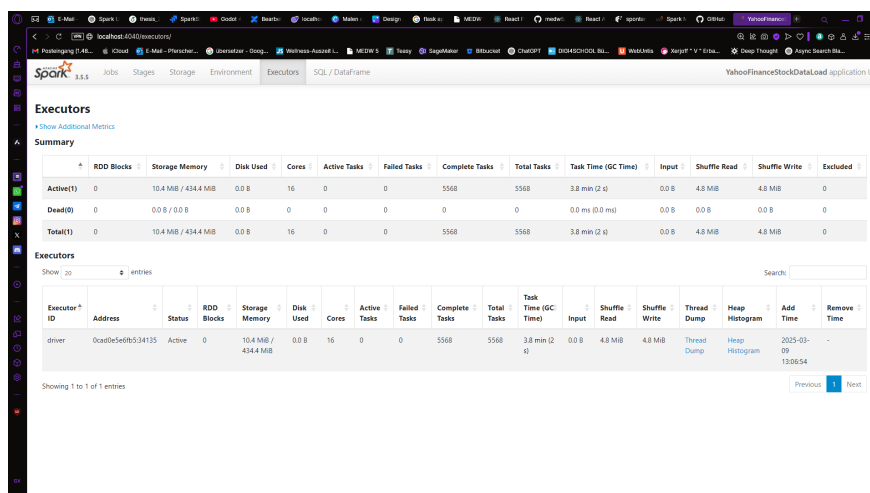
Abbildung 6.2.: Spark UI bei 30 Wertpapieren

60 Wertpapiere:



CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
0cad9e5e6fb5	python-app	743.66%	2.187GiB / 7.623GiB	27.64%	15.8MB / 31.7MB	0B / 0B	444
f49d3b6c37c2	tradingbotroot-db-1	11.17%	491.5MiB / 7.623GiB	6.39%	79.9MB / 34.7MB	0B / 0B	40

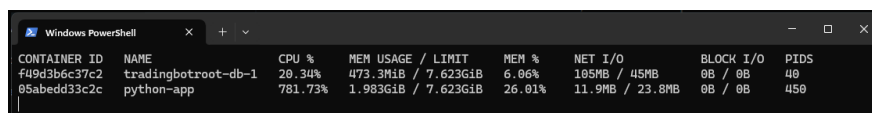
Abbildung 6.3.: Docker Stats bei 60 Wertpapieren



	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(1)	0	10.4 MB / 434.4 MB	0.0 B	16	0	0	5568	5568	3.8 min (2 s)	0.0 B	4.8 MB	4.8 MB	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(1)	0	10.4 MB / 434.4 MB	0.0 B	16	0	0	5568	5568	3.8 min (2 s)	0.0 B	4.8 MB	4.8 MB	0

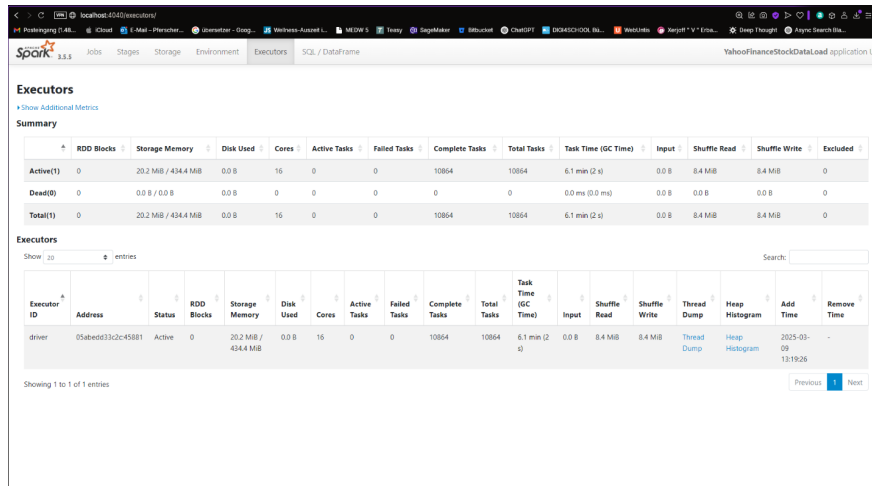
Abbildung 6.4.: Spark UI bei 60 Wertpapieren

120 Wertpapiere:



CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
f49d3b6c37c2	tradingbotroot-db-1	20.34%	473.3MiB / 7.623GiB	6.06%	105MB / 45MB	0B / 0B	40
05abedd33c2c	python-app	781.73%	1.983GiB / 7.623GiB	26.01%	11.9MB / 23.8MB	0B / 0B	450

Abbildung 6.5.: Docker Stats bei 120 Wertpapieren



Executors Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(1)	0	20.2 MB / 434.4 MB	0.0 B	16	0	0	10864	10864	6.1 min (2 s)	0.0 B	8.4 MB	8.4 MB	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(1)	0	20.2 MB / 434.4 MB	0.0 B	16	0	0	10864	10864	6.1 min (2 s)	0.0 B	8.4 MB	8.4 MB	0

Executors

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Thread Dump	Heap Histogram	Add Time	Remove Time
driver	05abed332c45881	Active	0	20.2 MB / 434.4 MB	0.0 B	16	0	0	10864	10864	6.1 min (2 s)	0.0 B	8.4 MB	8.4 MB	Thread Dump	Heap Histogram	2025-01-09 13:19:26	-

Abbildung 6.6.: Spark UI bei 120 Wertpapieren

6.2. Effizienz und Genauigkeit der Strategien

6.2.1. Vergleich der Strategien hinsichtlich Trefferquote(Hit-Rate)

Folglich wird jede Strategie hinsichtlich ihrer Hit-Rate über alle 12 Aktien bewertet.

Strategie 1:

- Anzahl der Trades: 48
- Hit-Rate: 68.75%

Strategie 2:

- Anzahl der Trades: 9
- Hit-Rate: 77.78%

Strategie 3:

- Anzahl der Trades: 59
- Hit-Rate: 69.49%

Aufgrund der 6.9-fachen Anzahl der Trades aus Strategie 2 und der nur um 8.29% niedrigeren Hitrate ist **Strategie 3** die profitabelste Strategie.

6.2.2. Vergleich der durch die Regressionsanalyse ermittelte ausschlaggebendste Indikator

Nachstehend wird der ausschlaggebendste technische Indikator, der mit dem Koeffizientenvergleich der Regressionsanalyse ermittelt wurde, verglichen. **Strategie 1:**

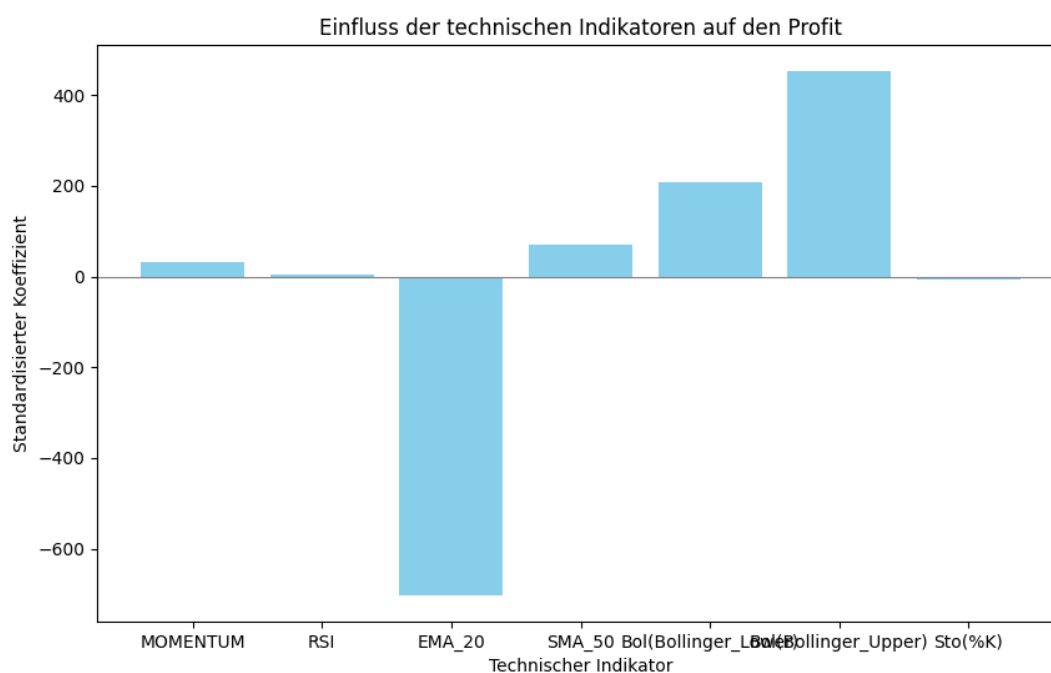


Abbildung 6.7.: Strategie_1_Statistik

Der ausschlaggebendste technische Indikator von Strategie 1 ist der EMA_20 mit einem Korrelationskoeffizienten von -703.183.

Strategie 2:

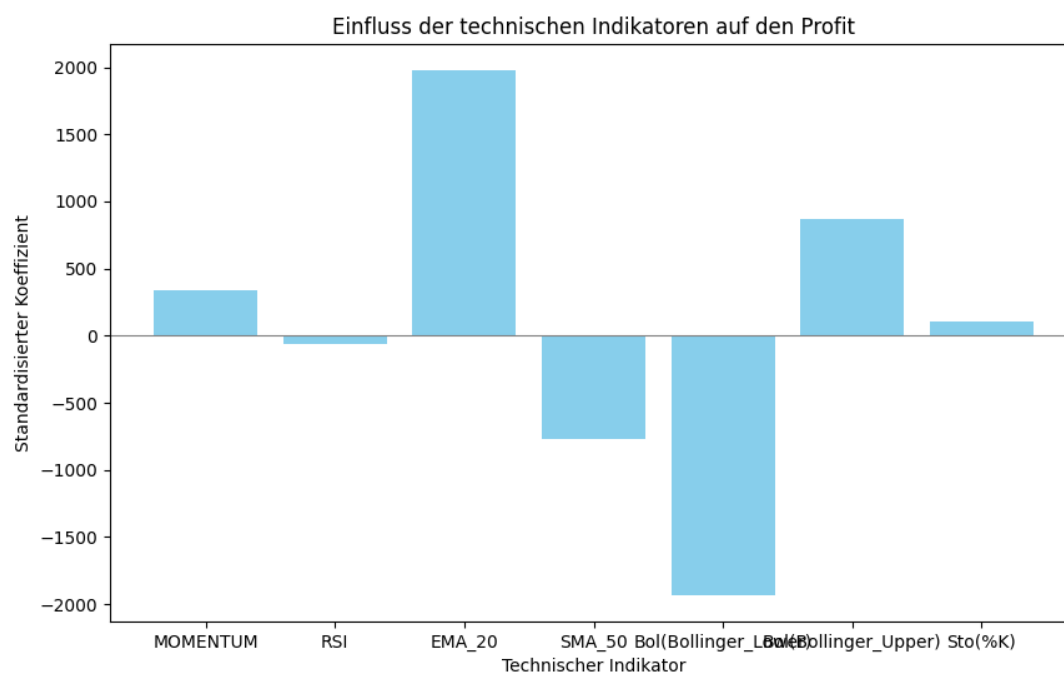


Abbildung 6.8.: Strategie_2_Statistik

Der ausschlaggebendste technische Indikator von Strategie 2 ist der EMA_20 mit einem Korrelationskoeffizienten von 1977.292.

Strategie 3:

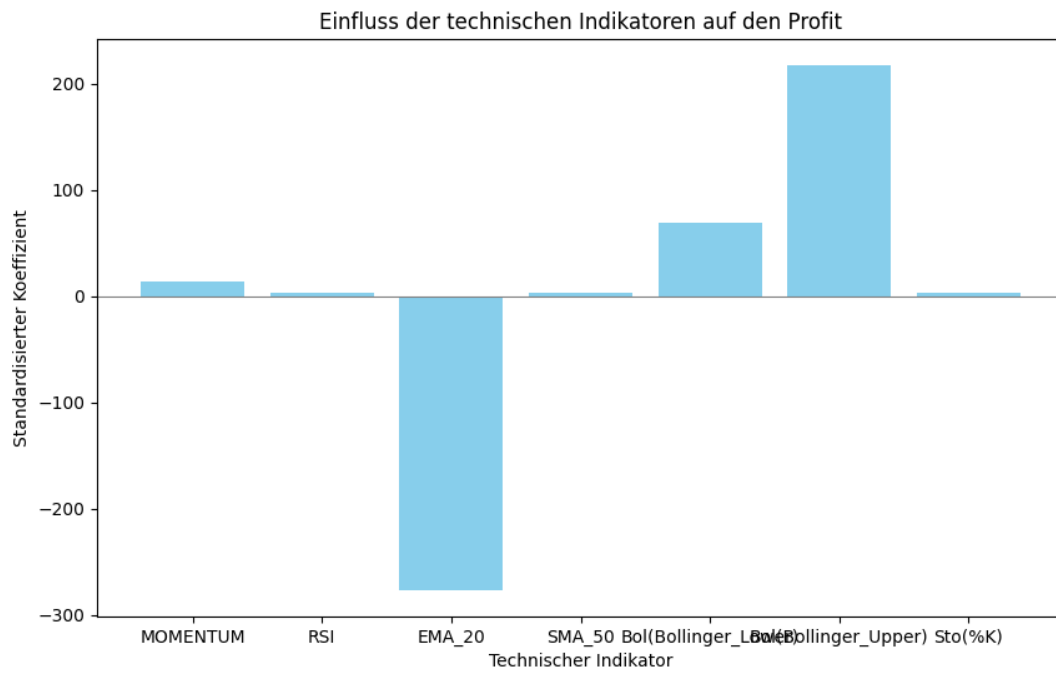


Abbildung 6.9.: Strategie_3_Statistik

Der ausschlaggebendste technische Indikator von Strategie 3 ist der EMA_20 mit einem Korrelationskoeffizienten von -276.822.

Im Zuge der Ergebnisse wurde bei allen 3 Strategien der EMA_20 als ausschlaggebenster Indikator ermittelt.

7. Prototyp Architektur

7.1. Systemarchitektur

7.1.1. Kommunikationsprotokolle und Systemarchitektur

Systemarchitektur Übersicht

Das System basiert auf einer verteilten Architektur, die auf Docker-Containern läuft. Diese Container beinhalten verschiedene Komponenten, die miteinander über definierte Kommunikationsprotokolle interagieren. Die Hauptkomponenten sind:

1. Blazor Frontend .NET (Docker Container für Weboberfläche)

- Dient als Benutzeroberfläche, über die Nutzer Daten einsehen und mit dem System interagieren können.
- Kommuniziert direkt mit der MySQL-Datenbank, um Finanzdaten abzurufen.
- Ermöglicht Benutzern die Verwaltung ihres Depots und das Anzeigen von Handelsentscheidungen.

2. Flask API (Docker Container für Handelsentscheidungen)

- Hat die alleinige Aufgabe, Handelsentscheidungen zu berechnen.
- Führt ein Skript aus, das basierend auf aktuellen Finanzdaten eine Investitionsempfehlung gibt.
- Kommuniziert nicht direkt mit MySQL oder Blazor, sondern wird nur durch externe Anfragen (z. B. vom Benutzer) gestartet.

3. Apache Spark (Docker Container für Datenverarbeitung)

- Ruft historische Finanzdaten über Yahoo Finance API (yfinance) ab.
- Bereitet große Mengen an Finanzdaten auf und speichert sie in MySQL.
- Wird regelmäßig oder bei Bedarf gestartet, um neue Daten in die Datenbank einzufügen.
- Kommuniziert ausschließlich mit Yahoo Finance und MySQL, aber nicht direkt mit Blazor oder Flask.

4. MySQL Datenbank (Docker Container für Speicherung von Finanzdaten)

- Speichert historische Marktdaten, technische Indikatoren und Benutzerdaten.

- Wird von Apache Spark mit neuen Finanzdaten gefüllt.
- Das Blazor-Frontend ruft Daten direkt aus der MySQL-Datenbank ab, um sie anzuzeigen.

7.1.2. Prozesskommunikation

Kommunikationsprotokolle zwischen den Komponenten

Die einzelnen Container kommunizieren über standardisierte Protokolle miteinander:

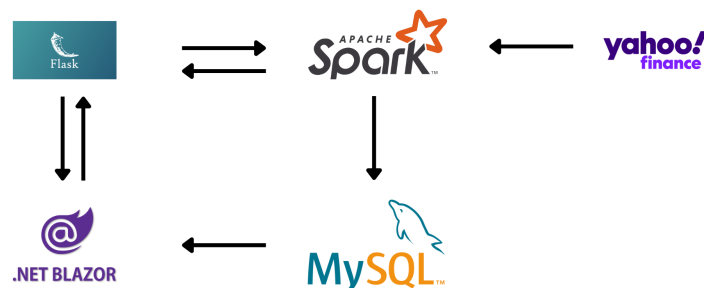


Abbildung 7.1.: Prototyp Architektur

1. Blazor - MySQL

Protokoll: SQL (JDBC / MySQL Connector)

Beschreibung: Blazor kommuniziert direkt mit der MySQL-Datenbank, um Benutzeranfragen zu verarbeiten.


Beispiel: Ein Benutzer ruft sein Portfolio auf → Blazor sendet eine SELECT-Abfrage an MySQL. Die Daten werden zurückgegeben und auf der Webseite angezeigt.

2. Flask API - Apache Spark

Protokoll: Python Skript-Aufruf

Beschreibung: Die Flask API wird bei Bedarf aufgerufen, um eine Handelsentscheidung zu treffen. Sie ruft kein externes System auf, sondern führt ein internes Python-Skript aus, das eine Analyse basierend auf den bereits in MySQL gespeicherten Daten durchführt.

Beispiel: Ein Benutzer fragt eine neue Investitionsempfehlung an → Die Flask API startet das Python-Skript für die Analyse. Das Skript verarbeitet historische Marktdaten aus der MySQL-Datenbank und trifft eine Handelsentscheidung.

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

3. Apache Spark - Yahoo Finance

Protokoll: HTTP (über yfinance API)

Beschreibung: Apache Spark lädt historische Marktdaten von Yahoo Finance. Die Daten werden über API-Requests basierend auf Ticker-Symbolen, Zeiträumen und Intervallen abgerufen.

Beispiel: Spark ruft AAPL-Daten für die letzten 6 Monate ab. Diese Daten werden anschließend für Analysen in MySQL gespeichert.

4. Apache Spark - MySQL

Protokoll: SQL (JDBC / MySQL Connector)

Beschreibung: Apache Spark sendet SQL-Queries an MySQL, um historische Daten und technische Indikatoren zu speichern.

Beispiel: Nach dem Abruf von Yahoo Finance speichert Spark neue Kursdaten mit INSERT INTO-Statements in MySQL. Spark kann auch große Datenmengen parallel verarbeiten und direkt in die Datenbank schreiben.

7.1.3. Verwendete Libraries zur Datenverarbeitung

1. Pandas (35)

Pandas ist eine leistungsstarke Open-Source-Bibliothek für die Datenanalyse in Python. Sie bietet Datenstrukturen und Funktionen zur effizienten Manipulation numerischer Tabellen und Zeitreihen.


Verwendung im Skript

```

1 start_dates = {
2     "1h": (pd.Timestamp(end_date) -
3           ↪ pd.DateOffset(months=6)).strftime("%Y-%m-%d"),
4     "1d": (pd.Timestamp(end_date) -
5           ↪ pd.DateOffset(years=1)).strftime("%Y-%m-%d")
6 }
```

pandas (pd) wird hier verwendet, um Zeitstempel zu manipulieren.

2. yfinance (36)

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

yfinance ist eine Python-Bibliothek, die den Zugriff auf Finanzdaten von Yahoo Finance ermöglicht. Sie erlaubt das Abrufen von historischen Marktdaten, Aktienkursen und weiteren Finanzinformationen.

Verwendung im Skript

```

1 ticker = yf.Ticker(self.ticker_symbol)
2 self.historical_data = ticker.history(start=self.start_date,
  ↳ end=self.end_date, interval=self.interval)

```

yfinance (yf) wird hier verwendet, um historische Finanzdaten abzurufen.

3. PyMySQL (37)

PyMySQL ist eine reine Python-Implementierung des MySQL-Client-Servers-Protokolls. Es ermöglicht die Verbindung und Interaktion mit MySQL-Datenbanken.

Verwendung im Skript

```

1 connection = pymysql.connect(
2     host='db',
3     user='root',
4     password='123mysql',
5     database='acatradinbotdb'
6 )

```

pymysql wird hier verwendet, um eine Verbindung mit der Datenbank herzustellen.

4. PySpark (38)

PySpark ist die Python-API für Apache Spark, ein Framework für die Verarbeitung großer Datenmengen. Es ermöglicht verteilte Datenverarbeitung und In-Memory-Computing.

Verwendung im Skript

```

1 spark = SparkSession.builder \
2     .appName("YahooFinanceStockDataLoad") \
3     .master("local[*]") \
4     .getOrCreate()

```

Hier wird eine Spark-Session gestartet, die als Grundlage für die Verarbeitung großer Datenmengen dient. local[*] nutzt alle verfügbaren CPU-Kerne für die Berechnung.

7.2. Relationale Datenbank im finanzspezifischen Kontext

7.2.1. Speichieranforderungen in MySQL

Die Speicherung von Finanzdaten in einer relationalen Datenbank wie MySQL stellt hohe Anforderungen an Effizienz, Performance und Skalierbarkeit. Finanzdaten umfassen riesige Mengen an historischen Preisdaten, Handelsvolumen, technischen Indikatoren und Benutzerinformationen, die optimal verwaltet werden müssen.

Speichieranforderungen für Finanzdaten

- **Hohe Datenvolumina:** Historische Marktdaten wachsen kontinuierlich an, da für jedes Wertpapier in verschiedenen Zeitintervallen Daten gespeichert werden.
- **Zeitkritische Abfragen:** Handelsalgorithmen benötigen schnelle Datenbankabfragen, um rechtzeitig Entscheidungen treffen zu können.
- **Indexierung und Partitionierung:** Da Finanzdaten oft nach Datum oder Ticker-Symbol abgefragt werden, ist eine effiziente Indexierung erforderlich.

7.2.2. Datenkonsistenz und -sicherheit

Datenkonsistenz in relationalen Datenbanken

Datenkonsistenz bezeichnet die Korrektheit, Vollständigkeit und Zuverlässigkeit von gespeicherten Informationen innerhalb eines Datenbanksystems. In einer relationalen Datenbank wie MySQL bedeutet Konsistenz, dass alle gespeicherten Daten den vordefinierten Integritätsregeln entsprechen. Dazu gehören beispielsweise die Einhaltung von Fremdschlüsselbeziehungen, referenzieller Integrität und bestimmten Einschränkungen, die verhindern, dass falsche oder unvollständige Daten gespeichert werden.

Datenkonsistenz ist essenziell, wenn mehrere Prozesse gleichzeitig auf eine Datenbank zugreifen und Änderungen durchführen. Ohne geeignete Mechanismen könnte es zu Inkonsistenzen kommen, beispielsweise durch unvollständig verarbeitete Transaktionen oder konkurrierende Schreibvorgänge.

Relevanz der Datenkonsistenz im Finanzbereich

Im Finanzsektor sind exakte und verlässliche Daten besonders wichtig, da bereits kleine Unstimmigkeiten gravierende Konsequenzen haben können. Finanzsysteme verarbeiten riesige Mengen an Marktdaten, die für Handelsentscheidungen, Portfoliomanagement und Risikobewertungen genutzt werden. Fehlerhafte oder inkonsistente Daten könnten zu falschen Handelsstrategien, fehlerhaften Kontoständen oder sogar zu regulatorischen Problemen führen.

8. Fazit und Ausblick

8.1. Zusammenfassung der Ergebnisse im Bezug auf Datenverarbeitung und Regressionsmodelle

8.1.1. Schlussfolgerungen aus den Performance-Tests bei der Datenverarbeitung

Die durchgeführten Messungen haben gezeigt, dass Apache Spark auch bei steigenden Datenmengen eine robuste und weitgehend lineare Skalierung aufweist. Trotz einer Verdopplung oder Vervierfachung der Wertpapieranzahl blieb die Verarbeitungsgeschwindigkeit auf einem effizienten Niveau, ohne dass gravierende Engpässe in CPU- oder Speicherauslastung auftraten.

Die Ergebnisse verdeutlichen, dass sich Spark für datenintensive Anwendungen wie algorithmisches Trading oder Regressionsanalysen in hohem Maße eignet und durch gezielte Optimierungsmaßnahmen (z.B. Partitionsanpassungen, Caching, Clusterbetrieb) weiter beschleunigt werden kann.

8.1.2. Relevanz der Regressionsanalyse für Finanzvorhersagen

Die Regressionsanalyse ist ein wichtiges Werkzeug zur Identifikation von Mustern und Zusammenhängen in der Finanzwelt. Sie hilft bei der Entwicklung von Vorhersagemodellen und beim allgemeinen Verständnis von Trends. Allerdings stößt sie an ihre Grenzen, da Finanzmärkte oft von unvorhersehbaren Ereignissen beeinflusst werden.

In Zukunft werden hybride Ansätze, die klassische Regressionsmodelle mit Methoden des maschinellen Lernens kombinieren, immer wichtiger. Sie könnten Prognosen präziser und flexibler machen, indem sie komplexe Muster erkennen, die traditionelle Modelle übersehen. Die Herausforderung besteht darin, diese Methoden weiterzuentwickeln und an die sich ständig verändernden Finanzmärkte anzupassen.

9. Projektdokumentation

9.1. Arbeitsteilung

Florian Brunner:


- Technische Indikatoren
- Handelsstrategien in Python
- Depot Übersicht, Profil Seite

Karl Pferscher:

- Laden von Historischen Marktdaten
- Flask API
- Restliches Frontend (Dashboard, History, Depot-Detail, Discover, Aktien-Detail)

9.2. Stundenliste

9.2.1. Florian Brunner:

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT		
	Krems		
	Abteilung:	Informationstechnologie	
	Ausbildungsschwerpunkt:		

Name	Arbeitspaket	Datum	Arbeitsstunden
Florian Brunner	A1.1	20.09.2024	5
Florian Brunner	A1.2	26.09.2024	4
Florian Brunner	A1.2	26.09.2024	4
Florian Brunner	A2.1.1	28.09.2024	3
Florian Brunner	A2.1.1	29.09.2024	4
Florian Brunner	A7.2	30.09.2024	3
Florian Brunner	A2.1.2	01.09.2024	2
Florian Brunner	A2.2.1	03.10.2024	4
Florian Brunner	A2.2.2	04.10.2024	2
Florian Brunner	A4.1.1	09.10.2024	2
Florian Brunner	A7.2	10.10.2024	3
Florian Brunner	A4.1.1	11.10.2024	2
Florian Brunner	A2.2.2	11.10.2024	1
Florian Brunner	A4.1.1	15.10.2024	1
Florian Brunner	A4.1.2	20.10.2024	3
Florian Brunner	A2.2.2	24.10.2024	3
Florian Brunner	A4.1.1	26.10.2024	1
Florian Brunner	A7.2	28.10.2024	4
Florian Brunner	A4.1.1	30.10.2024	3
Florian Brunner	A4.1.2	02.11.2024	2
Florian Brunner	A2.2.2	06.11.2024	4
Florian Brunner	A4.1.1	08.11.2024	1
Florian Brunner	A4.1.2	12.11.2024	4
Florian Brunner	A4.1.1	14.11.2024	2
Florian Brunner	A4.1.3	14.11.2024	1
Florian Brunner	A4.1.3	15.11.2024	3
Florian Brunner	A7.2	17.11.2024	2
Florian Brunner	A4.1.3	20.11.2024	3
Florian Brunner	A4.1.3	23.11.2024	3
Florian Brunner	A4.2.1	24.11.2024	2
Florian Brunner	A4.2.1	25.11.2024	3
Florian Brunner	A4.2.1	26.Nov	3
Florian Brunner	A4.1.3	28.11.2024	2
Florian Brunner	A4.2.1	01.12.2024	3
Florian Brunner	A4.2.1	03.12.2024	1
Florian Brunner	A4.2.1	08.12.2024	3
Florian Brunner	A4.2.2	11.12.2024	3
Florian Brunner	A4.1.3	18.11.2024	2
Florian Brunner	A4.2.2	19.11.2024	3
Florian Brunner	A4.2.2	23.11.2024	4
Florian Brunner	A4.2.2	24.11.2024	2
Florian Brunner	A4.3.1	24.11.2024	1
Florian Brunner	A3.3.3	25.11.2024	3
Florian Brunner	A3.3.3	26.11.2024	2
Florian Brunner	A3.3.8	29.11.2024	4
Florian Brunner	A4.3.1	02.12.2024	3

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT		
	Krems		
	Abteilung:	Informationstechnologie	
	Ausbildungsschwerpunkt:		


Florian Brunner	A4.3.1	03.12.2024	2
Florian Brunner	A4.3.1	04.12.2024	1
Florian Brunner	A3.3.8	07.12.2024	3
Florian Brunner	A3.3.3	09.12.2024	2
Florian Brunner	A4.3.2	11.12.2024	3
Florian Brunner	A4.3.3	17.12.2024	2
Florian Brunner	A3.3.8	21.12.2024	1
Florian Brunner	A4.3.2	28.12.2024	3
Florian Brunner	A4.3.2	29.12.2024	2
Florian Brunner	A4.3.2	30.12.2024	2
Florian Brunner	A4.3.3	03.01.2025	3
Florian Brunner	A4.3.3	08.01.2025	2
Florian Brunner	A4.3.3	14.01.2025	2
Florian Brunner	A4.3.2	18.01.2025	1
Florian Brunner	A4.2.3	19.01.2025	2
Florian Brunner	A4.2.3	22.01.2025	4
Florian Brunner	A6.1.1	23.01.2025	3
Florian Brunner	A6.1.1	25.01.2025	2
Florian Brunner	A6.1.1	27.01.2025	4
Florian Brunner	A7.1	28.01.2025	3
Florian Brunner	A7.1	29.01.2025	4
Florian Brunner	A7.2	30.01.2025	2
Florian Brunner	A7.1	02.01.2025	5
Florian Brunner	A7.1	06.01.2025	2
Florian Brunner	A7.1	08.01.2025	1
Florian Brunner	A7.1	05.01.2025	2
Florian Brunner	A7.1	10.01.2025	1
Florian Brunner	A7.2	20.01.2025	2
Florian Brunner	A7.1	28.01.2025	1
Florian Brunner	A7.1	03.02.2025	2
Florian Brunner	A7.1	10.02.2025	3
Florian Brunner	A7.1	15.02.2025	2
Florian Brunner	A7.1	23.02.2025	3
Florian Brunner	A7.1	27.02.2025	1


Arbeitspaket	Tatsächlich:	Erwartet:
1.1:	5	4
1.2:	8	8
2.1.1:	7	5
2.1.2:	2	2
2.2.1:	4	4
2.2.2:	10	6
3.3.3:	7	14
3.3.8:	8	10
4.1.1:	12	10
4.1.2:	9	7
4.1.3:	14	15
4.2.1:	15	15
4.2.2:	12	10
4.2.3:	3	6
4.3.1:	7	6
4.3.2:	11	12
4.3.3:	9	5
6.1.1:	9	10
7.1:	30	30
7.2:	16	10


Gesamtstunden:


201

Legende

 Tatsächliche Stunden > Erwartete Stunden

 Tatsächliche Stunden < Erwartete Stunden


 Tatsächliche Stunden = Erwartete Stunden

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:


9.2.2. Karl Pferscher:

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT		
	Krems		
	Abteilung:	Informationstechnologie	
	Ausbildungsschwerpunkt:		

Name	Arbeitspaket	Datum	Arbeitsstunden
Karl Pferscher	A1.1	20.09.2024	5
Karl Pferscher	A1.3	23.09.2024	3
Karl Pferscher	A1.3	25.09.2024	3
Karl Pferscher	A2.3	28.09.2024	4
Karl Pferscher	A2.3	29.09.2024	4
Karl Pferscher	A2.3	30.09.2024	4
Karl Pferscher	A2.3.1	02.10.2024	3
Karl Pferscher	A2.3.2	04.10.2024	2
Karl Pferscher	A2.3.2	10.10.2024	4
Karl Pferscher	A2.3.2	11.10.2024	2
Karl Pferscher	A2.3.3	11.10.2024	2
Karl Pferscher	A2.3.3	13.10.2024	5
Karl Pferscher	A3.1.1	20.10.2024	4
Karl Pferscher	A3.1.2	21.10.2024	3
Karl Pferscher	A3.1.2	22.10.2024	4
Karl Pferscher	A3.1.2	24.10.2024	2
Karl Pferscher	A3.1.2	25.10.2024	4
Karl Pferscher	A3.1.2	26.10.2024	2
Karl Pferscher	A3.2	27.10.2024	1
Karl Pferscher	A3.3.1	01.11.2024	5
Karl Pferscher	A3.3.1	02.11.2024	3
Karl Pferscher	A5.1.1	02.11.2024	2
Karl Pferscher	A5.1.2	03.11.2024	2
Karl Pferscher	A3.3.4	03.11.2024	2
Karl Pferscher	A5.1.2	04.11.2024	1
Karl Pferscher	A3.3.4	04.11.2024	4
Karl Pferscher	A3.3.4	06.11.2024	4
Karl Pferscher	A5.1.2	07.11.2024	1
Karl Pferscher	A3.3.5	09.11.2024	3
Karl Pferscher	A5.1.2	09.11.2024	1
Karl Pferscher	A3.3.5	13.11.2024	3
Karl Pferscher	A5.1.2	13.11.2024	2
Karl Pferscher	A3.3.5	15.11.2024	5
Karl Pferscher	A5.1.2	16.11.2024	3
Karl Pferscher	A3.3.5	18.11.2024	4
Karl Pferscher	A3.3.6	24.11.2024	3
Karl Pferscher	A3.3.6	28.11.2024	3
Karl Pferscher	A3.3.6	30.11.2024	2
Karl Pferscher	A5.1.2	30.11.2024	2
Karl Pferscher	A3.3.7	02.12.2024	3
Karl Pferscher	A3.3.7	06.12.2024	2
Karl Pferscher	A5.1.2	06.12.2024	3
Karl Pferscher	A3.3.7	07.12.2024	2
Karl Pferscher	A3.3.9	10.12.2024	3
Karl Pferscher	A3.3.9	12.12.2024	1
Karl Pferscher	A3.3.9	15.12.2024	2
Karl Pferscher			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems		
	Abteilung: Ausbildungsschwerpunkt:	Informationstechnologie	

Karl Pferscher	A5.1.2	19.12.2024	2
Karl Pferscher	A3.3.9	22.12.2024	3
Karl Pferscher	A5.1.2	23.12.2024	1
Karl Pferscher	A3.3.9	28.12.2024	1
Karl Pferscher	A5.1.1	29.12.2024	3
Karl Pferscher	A5.1.1	03.01.2025	3
Karl Pferscher	A5.1.2	04.01.2025	3
Karl Pferscher	A3.3.7	05.01.2025	2
Karl Pferscher	A3.3.6	07.01.2025	2
Karl Pferscher	A5.2.2	14.01.2025	4
Karl Pferscher	A5.2.1	15.01.2025	3
Karl Pferscher	A5.2.1	17.01.2025	3
Karl Pferscher	A5.2.2	20.01.2025	3
Karl Pferscher	A6.2	21.01.2025	3
Karl Pferscher	A6.2	23.01.2025	2
Karl Pferscher	A6.2	25.01.2025	3
Karl Pferscher	A6.2.1	26.01.2025	2
Karl Pferscher	A7.1	28.01.2025	3
Karl Pferscher	A7.1	29.01.2025	3
Karl Pferscher	A7.1	31.01.2025	3
Karl Pferscher	A7.1	02.02.2025	3
Karl Pferscher	A7.1	04.02.2025	4
Karl Pferscher	A7.1	05.02.2025	4
Karl Pferscher	A7.1	07.02.2025	4
Karl Pferscher	A7.2	15.02.2025	3
Karl Pferscher	A7.2	17.02.2025	4
Karl Pferscher	A7.2	20.02.2025	4

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaket	Tatsächlich:	Erwartet:
1.1:	5	4
1.3:	6	5
2.3:	12	12
2.3.1:	3	3
2.3.2:	8	9
2.3.3:	7	10
3.1.1:	4	3
3.1.2:	15	14
3.2:	1	2
3.3.1:	8	8
3.3.4:	10	14
3.3.5:	15	14
3.3.6:	10	9
3.3.7:	10	11
3.3.9:	10	11
5.1.1:	8	8
5.1.2:	21	18
5.2.1:	6	5
5.2.2:	7	5
6.2:	8	13
6.2.1:	2	3
7.1:	24	30
7.2:	11	10


Gesamtstunden:

210

Legende


 Tatsächliche Stunden > Erwartete Stunden

 Tatsächliche Stunden < Erwartete Stunden

 Tatsächliche Stunden = Erwartete Stunden

I. Literaturverzeichnis

- [1] Wikipedia-Autoren: *Big Data*. https://de.wikipedia.org/wiki/Big_Data.
- [2] Bolthoef, Eva: *Datengetriebenes Finanzmanagement: Potenzial und Herausforderungen von Big Data für die Finanzbranche*. <https://qonto.com/de/blog/news/business/datengetriebenes-finanzmanagement-wie-big-data-die-finanzbranche-optimieren-kann>.
- [3] FasterCapital: *Die Rolle von Big Data in der Finanzbranche*. <https://fastercapital.com/de/inhalt/Die-Macht-von-Big-Data-in-der-Finanzanalyse-nutzen.html#Einf-hrung-in-Big-Data-in-der-Finanzanalyse>.
- [4] N.A.D.R.: *Die Risiken und Chancen des algorithmischen Handels*. <https://www.nadr.de/aktuelles/wirtschaft/die-risiken-und-chancen-des-algorithmischen-handels/>.
- [5] Schulz, Marcus: *Vorteile des algorithmischen Trdings(24 Verfügbarkeit)*. <https://volume-trader.com/blog/der-einfluss-von-algorithmen-auf-moderne-trading-strategien-chancen-und-risiken>.
- [6] FasterCapital: *Algorithmischer Handel Die Leistungsfähigkeit des algorithmischen Handels im Big Data Zeitalter nutzen*. <https://fastercapital.com/de/inhalt/Algorithmischer-Handel--Die-Leistungsfähigkeit-des-algorithmischen-Handels-im-Bi.html>.
- [7] Schroer, Kevin: *Lineare Regression*. <https://www.bwl-lexikon.de/wiki/lineare-regression/>.
- [8] Talend: *Big Data im Finanzwesen: Ihr Guide für die Finanzdaten-Analyse*. <https://www.talend.com/de/resources/big-data-im-finanzwesen/>.
- [9] Kidd, Chrissy: *Distributed Systems: So funktionieren verteilte Systeme*. www.splunk.com/de_de/data-insider/what-are-distributed-systems.html.
- [10] Storage, Pure: *Was ist Distributed Data Processing?* <https://www.purestorage.com/de/knowledge/what-is-distributed-data-processing.html>.
- [11] Rau, Sebastian: *Algo Trading – So funktioniert Trading mit Algorithmen*. <https://finanzwissen.de/trading/strategien/algorithmisch/>.
- [12] Finanzportale, ADT: *Definition und Bedeutung von Technischen Indikatoren*. <https://boersenlexikon.net/technische-indikatoren/>.
- [13] Murphy, Casey: *Indikatoren-Popularität*. <https://www.investopedia.com/ask/answers/05/selffulfillingprophecy.asp>.
- [14] GmbH, CMC Markets Germany: *Einfacher Gleitender Durchschnitt(SMA)*. <https://www.cmcmarkets.com/de-at/hilfe/glossar/s/sma-indikator>.
- [15] Jennifer Shirin, Simon Rieger: *Simple Moving Average (SMA): Trading, Interpretation und Anwendung*. <https://bitcoin-2go.de/trading/indikatoren/sma/>.

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

- [16] GmbH, CMC Markets Germany: *Exponentieller Gleitender Durchschnitt (EMA)*. <https://www.cmcmarkets.com/de-at/hilfe/glossar/e/exponentieller-gleitender-durchschnitt>.
- [17] GmbH, CMC Markets Germany: *Relativer Stärke-Index (RSI)*. <https://www.cmcmarkets.com/de-at/hilfe/glossar/r/rsi-indikator>.
- [18] GmbH, CMC Markets Germany: *Stochastik (Stoch)*. <https://www.cmcmarkets.com/de-at/hilfe/glossar/s/stochastik>.
- [19] Thomas Müller, TM BÖRSENVERLAG AG: *Das GROSSE Buch der TECHNISCHEN INDIKATOREN: Momentum Erklärung – Technische Analyse*. <https://www.boerse.de/technische-indikatoren/Momentum-23>.
- [20] GmbH, CMC Markets Germany: *Bollinger Bänder*. <https://www.cmcmarkets.com/de-at/hilfe/glossar/b/bollinger-baender>.
- [21] JonskiC, KW, ShiningSeraph1m: *Lineare Regression*. https://de.wikipedia.org/wiki/Lineare_Regression.
- [22] JonskiC, KW, ShiningSeraph1m: *Lineare Regression*. https://de.wikipedia.org/wiki/Lineare_Regression.
- [23] contributors, Wikipedia: *Regressionsparameter*. <https://de.wikipedia.org/wiki/Regressionsparameter>.
- [24] Analytics-Team, Astera: *Skalierbare ETL-Architekturen: Umgang mit großen Datenmengen*. <https://www.astera.com/de/knowledge-center/scalable-etl-architectures/>.
- [25] contributors, Wikipedia: *Analytisches Informationssystem*. https://de.wikipedia.org/wiki/Analytisches_Informationssystem.
- [26] SnapLogic: *Integrationsanforderungsdokument - Erläuterung und Überblick*. <https://www.snaplogic.com/de/glossary/integration-requirements-document>.
- [27] contributors, Wikipedia: *Interoperabilität*. <https://de.wikipedia.org/wiki/Interoperabilität>.
- [28] Schneppat, Jörg Owe: *Apache Spark*. <https://gpt5.blog/apache-spark/>.
- [29] Microsoft: *Grundlegende Konzepte für Apache Spark in Azure Synapse Analytics*. <https://learn.microsoft.com/de-de/azure/synapse-analytics/spark/apache-spark-concepts>.
- [30] contributors, Wikipedia: *Apache Spark*. https://de.wikipedia.org/wiki/Apache_Spark.
- [31] Wuttke, Laurenz: *ETL mit Apache Spark*. <https://datasolut.com/etl-mit-spark/>.
- [32] AWS: *Was ist Batch-Verarbeitung?* <https://aws.amazon.com/de/what-is/batch-processing/#:~:text=Batch%20processing%20is%20the%20method,run%20on%20individual%20data%20transactions>.

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

- [33] Amit, Hey: *Apache Hadoop vs Spark*. <https://medium.com/%40heyamit10/apache-hadoop-vs-spark-f846060931c8>.
- [34] Simon04: *Python*. [https://de.wikipedia.org/wiki/Python_\(Programmiersprache\)](https://de.wikipedia.org/wiki/Python_(Programmiersprache)).
- [35] Klein, Bernd: *Pandas Einführung*. <https://www.python-kurs.eu/pandas.php>.
- [36] Singh, Prashant: *Fetching Stock Market Data Made Simple with Python and yFinance*. <https://www.linkedin.com/pulse/fetching-stock-market-data-made-simple-python-yfinance-prashant-singh-a2ysc#:~:text=yFinance%20is%20a%20Python%20library,complex%20APIs%2C%20no%20hidden%20costs>.
- [37] Geeksforgeeks: *How To Install Pymysql In Python?* <https://www.geeksforgeeks.org/how-to-install-pymysql-in-python/>.
- [38] Spark, Apache: *PySpark Overview*. <https://spark.apache.org/docs/latest/api/python/index.html>.
- [39] contributors, Wikipedia: *Command pattern*. https://en.wikipedia.org/wiki/Command_pattern.
- [40] Bundesbank, Deutsche: *Distributed-Ledger-Technologien im Zahlungsverkehr und in der Wertpapierabwicklung: Potenziale und Risiken*. www.bundesbank.de/resource/blob/665446/cfd6e8f8e0f2563b9fc1f48fabda8ca2/mL/2017-09-distributed-ledger-technologien-data.pdf.

II. Abbildungsverzeichnis


3.1. Verteilte Datenverarbeitung	20
3.2. SMA (50, 100, 200) Apple, 4h	23
3.3. EMA (50) Apple, 4h	24
3.4. RSI Apple, 4h	25
3.5. Stochastik Apple, 4h	27
3.6. Momentum Apple, 4h	28
3.7. Bollinger Bänder Apple, 4h	29
3.8. Lineare Regression	31
3.9. Apache Spark	35
3.10. Apache Spark vs. Hadoop	38
5.1. Verarbeitungszeit bei steigender Last	60
6.1. Docker Stats bei 30 Wertpapieren	67
6.2. Spark UI bei 30 Wertpapieren	67
6.3. Docker Stats bei 60 Wertpapieren	68
6.4. Spark UI bei 60 Wertpapieren	68
6.5. Docker Stats bei 120 Wertpapieren	68
6.6. Spark UI bei 120 Wertpapieren	69
6.7. Strategie_1_Statistik	70
6.8. Strategie_2_Statistik	71
6.9. Strategie_3_Statistik	72
7.1. Prototyp Architektur	74

III. Tabellenverzeichnis

2.1. Vor und Nachteile des algorithmischen Tradings (4; 5)	14
5.1. Veränderung der Verarbeitungszeit von Apache Spark	60
5.2. CPU- und Speicherauslastung bei steigender Anzahl an Wertpapieren	60
A.1. Kapitelverzeichnis	95


IV. List of Listings

4.1. SMA-Implementierung	40
4.2. EMA-Implementierung	41
4.3. Momentum-Implementierung	42
4.4. RSI-Implementierung	43
4.5. Bollinger Bänder-Implementierung	44
4.6. Stochastik-Implementierung	45
4.7. Regressionsmodell-Implementierung	46
4.8. Erstellung der Spark Session	47
4.9. Abrufen historischer Finanzdaten	48
4.10. Konvertierung der Daten in ein Spark DataFrame	50
4.11. Speicherung der Daten in MySQL	52
4.12. Erstellung der Spark-Session für die verteilte Verarbeitung	53
4.13. Abruf der gespeicherten Wertpapiere aus der MySQL-Datenbank	54
4.14. Abruf des zuletzt gespeicherten Kursdatums für ein Wertpapier	55
4.15. Abruf neuer Finanzmarktdaten über Yahoo Finance	56
4.16. Umwandlung der neuen Finanzmarktdaten in ein Spark DataFrame	57
5.1. Messung der Verarbeitungszeit mittels time.time()	58
5.2. Generelle Strategiestruktur	62
5.3. Parameter der Strategie 1	63
5.4. Parameter der Strategie 2	64
5.5. Parameter der Strategie 3	65

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

V. Abkürzungsverzeichnis

...


	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

A. Appendix

A.1. Kapitelverzeichnis

Kapitel	Autor
1	Karl Pferscher
2.1.1	Florian Brunner
2.1.2	Florian Brunner
2.2.1	Karl Pferscher
2.2.2	Florian Brunner
2.3.1	Florian Brunner
2.3.2	Florian Brunner
2.4.1	Florian Brunner
2.4.2	Karl Pferscher
3.1.1	Karl Pferscher
3.1.2	Karl Pferscher
3.2.1	Florian Brunner
3.2.2	Florian Brunner
3.3.1	Florian Brunner
3.3.2	Florian Brunner
3.4.1	Karl Pferscher
3.4.2	Karl Pferscher
3.5.1	Karl Pferscher
3.5.2	Karl Pferscher
3.5.3	Karl Pferscher
4.1.1	Florian Brunner
4.2.1	Florian Brunner
4.3.1	Karl Pferscher
4.3.2	Karl Pferscher
5.1.1	Karl Pferscher
5.1.2	Karl Pferscher
5.1.3	Karl Pferscher
5.2.1	Florian Brunner
5.2.2	Florian Brunner
6.1.1	Karl Pferscher
6.2.1	Florian Brunner
6.2.2	Florian Brunner
7.1.1	Karl Pferscher
7.1.2	Karl Pferscher
7.1.3	Karl Pferscher
7.2.1	Karl Pferscher
7.2.2	Karl Pferscher
8.1.1	Karl Pferscher
8.1.2	Florian Brunner
9	Karl Pferscher

Tabelle A.1.: Kapitelverzeichnis

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

A.2. Besprechungsprotokolle

Betreuungsprotokoll zur Diplomarbeit
lfd. Nr.: 1

Themenstellung: ACA Tradingbot
 Kandidaten/Kandidatinnen: Karl Pferscher, Florian Brunner

Jahrgang: 5BHITM
 Betreuer/in: Panhofer Paul
 Ort: HTL Krems
 Datum: 01.10.2024
 Zeit: 10:00

Besprechungsinhalt:

Name	Notiz
Meeting mit dem Auftraggeber (Kickoffmeeting)	Besprechung des Projektablaufs und der Umsetzungsstrategie mit dem Auftraggeber und dem Betreuungslehrer.

Keine Einwände

Aufgaben:

Name	Notiz	zu erledigen bis
Florian Brunner	PSP und Pflichtenheft	27.11.2024
Karl Pferscher	Mockups und Arbeitspaketbeschreibung	27.11.2024

Betreuungsprotokoll zur Diplomarbeit

lfd. Nr.: 2

Themenstellung: ACA Tradingbot
 Kandidaten/Kandidatinnen: Karl Pferscher, Florian Brunner

Jahrgang: 5BHITM
 Betreuer/in: Panhofer Paul
 Ort: HTL Krems
 Datum: 21.10.2024
 Zeit: 07:00

Besprechungsinhalt:

Name	Notiz
Abgabe der Planungsdokumente der Diplomarbeit	<ul style="list-style-type: none"> Abgabe des Pflichtenheftes. Abgabe des Projektstrukturplans und der Arbeitspaketbeschreibung. Abgabe der Mockups.

Keine Einwände

Aufgaben:

Name	Notiz	zu erledigen bis
Karl Pferscher	Implementierung der Python Skripte: <ul style="list-style-type: none"> Erstellen der Inserts Laden der Historischen Marktdaten 	27.11.2024
Florian Brunner	Implementierung der Frontendseiten: <ul style="list-style-type: none"> Depotübersicht Profil 	27.11.2024

Betreuungsprotokoll zur Diplomarbeit

Ild. Nr.: 3

Themenstellung: ACATradingbot
 Kandidaten/Kandidatinnen: Karl Pferscher, Florian Brunner

Jahrgang: 5BHITM
 Betreuer/in: Panhofer Paul
 Ort: HTL-Krems
 Datum: 27.11.2024
 Zeit: 13:00

Besprechungsinhalt:

Name	Notiz
Implementierung	Implementierung der Python Skripte: <ul style="list-style-type: none"> • Erstellen der Inserts • Laden der Historischen Marktdaten Implementierung der Frontendseiten: <ul style="list-style-type: none"> • Depotübersicht • Profil

Aufgaben:

Name	Notiz	zu erledigen bis
Florian Brunner	<ul style="list-style-type: none"> • Implementierung der Trading Strategien • Implementierung der Technischen Indikatoren 	09.01.2024
Karl Pferscher	Implementierung des Frontend: <ul style="list-style-type: none"> • Dashboard • Depot-Detail-View • Titelaufstellung • Login • Registrierung 	09.01.2024

Betreuungsprotokoll zur Diplomarbeit
lfd. Nr.: 4

Themenstellung: ACA Tradingbot
 Kandidaten/Kandidatinnen: Karl Pferscher, Florian Brunner

Jahrgang: 5BHITM
 Betreuer/in: Panhofer Paul
 Ort: HTL Krems
 Datum: 10.12.2024
 Zeit: 09:30

Besprechungsinhalt:

Name	Notiz
Kontrolle der bisherigen Implementierungen	Implementierung des Frontend: <ul style="list-style-type: none"> • Dashboard • Depot-Detail-View • Titelauflistung • Login • Registrierung
	Implementierung der Trading Strategien Implementierung der Technischen Indikatoren

Keine Einwände

Aufgaben:

Name	Notiz	zu erledigen bis
Florian Brunner	<ul style="list-style-type: none"> • Implementierung eines Backtesting-moduls • Implementierung Regressionsmoduls • Weitere Strategieoptimierungen 	14.2.2024
	Fertigstellung des Frontend: <ul style="list-style-type: none"> • Titelanalyse Optimierung der Verteilten Datenverarbeitung	14.2.2024

Betreuungsprotokoll zur Diplomarbeit
lfd. Nr.: 5

Themenstellung: ACA Tradingbot
 Kandidaten/Kandidatinnen: Karl Pferscher, Florian Brunner

Jahrgang: 5BHITM
 Betreuer/in: Panhofer Paul
 Ort: HTL Krems
 Datum: 14.2.2024
 Zeit: 14:00


Besprechungsinhalt:


Name	Notiz
Code Freeze	<ul style="list-style-type: none"> • Implementierung eines Backtesting-moduls • Implementierung Regressionsmoduls • Weitere Strategieoptimierungen Fertigstellung des Frontend: <ul style="list-style-type: none"> • Titelanalyse • Optimierung der Verteilten Datenverarbeitung

Keine Einwände

Aufgaben:

Name	Notiz	zu erledigen bis

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

	HTL Krems Höhere Lehranstalt für Informationstechnologie Ausbildungsschwerpunkt	Reife- und Diplomprüfung
---	--	-------------------------------------

Abnahmeprotokoll zur Diplomarbeit


Themenstellung: ACA Tradingbot
 Kandidaten/Kandidatinnen: Karl Pferscher, Florian Brunner

 Jahrgang: 5BHITM
 Betreuer/in: Panhofer Paul
 Auftraggeber: Dipl.-Ing. Mag. Georg Ungerböck X-WORKS GmbH
 Ort: HTL Krems
 Datum: 19.2.2024
 Zeit: 15:30


Besprechungsinhalt:

Name	Notiz
Abnahme	<ul style="list-style-type: none"> • Präsentation sämtlicher Kernfunktionalitäten der Applikation • Positives Feedback und Bestätigung der Funktionsfähigkeit durch den Auftraggeber • Kurze Klärung offener Fragen mit direkten Antworten • Keine weiteren offenen Punkte oder Nachbesserungen festgestellt • Abnahme der Diplomarbeit durch den Auftraggeber bestätigt

Signiert von:	Georg Ungerböck
Datum:	25.03.2025 08:56:09
 <p>Dieses Dokument ist digital signiert! Dieses ist ein elektronisch signiertes Dokument. Signatur und Inhalt sind elektronisch signiert. Datum: 25.03.2025 08:56:09 Ort: HTL Krems</p>	

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

A.3. Pflichtenheft

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

Pflichtenheft: Tradingbot

1. Einleitung

Im Rahmen der Diplomarbeit soll ein Tradingbot entwickelt werden, der verschiedene Depots verwalten und für jedes Depot Kauf- und Verkaufsentscheidungen treffen kann. Ziel ist es, die Funktionen so umzusetzen, dass der Benutzer eine intuitive Weboberfläche zur Verwaltung der Depots nutzen kann. Die Kauf- und Verkaufsentscheidungen des Bots sollen auf Basis von Analysen der verfügbaren Titel im Depot getroffen werden. Die Aktionen des Bots sowie die Entscheidungen sollen in einer Datenbank gespeichert werden.

2. Zielsetzung und Produktbeschreibung

Das Ziel dieses Projekts ist die Implementierung eines voll funktionsfähigen Tradingbots, der mehrere Depots mit unterschiedlichen handelbaren Titeln (Aktien, ETFs, Forex, Kryptowährungen) verwalten und autonom Kauf- und Verkaufsentscheidungen treffen kann. Der Benutzer soll eine Weboberfläche zur Interaktion nutzen können, in der er Depots erstellen, Titel hinzufügen oder entfernen und die Historie der getätigten Aktionen einsehen kann. Alle Aktionen sollen in einer SQL-Datenbank gespeichert werden.


3. Funktionale Anforderungen

3.1 Depotmanagement

- **Erstellen eines Depots:**
 - Benutzer kann mehrere Depots anlegen.
 - Jedes Depot erhält einen eindeutigen Namen, den der Benutzer festlegt.
- **Zusammenstellung der handelbaren Titel:**
 - Auswahl von Aktien, ETFs, Fonds, Forex und Kryptowährungen.
 - Hinzufügen von Titeln aus einer vordefinierten Liste.
 - Löschen von Titeln aus einem Depot.
- **Einzelbetrachtung der Depots:**
 - Der Tradingbot analysiert jedes Depot separat und trifft Kauf- bzw. Verkaufsentscheidungen basierend auf den handelbaren Titeln in diesem Depot.

3.2 Analyse und Handelsentscheidungen

- **Analyse der Titel:**
 - Der Bot analysiert alle im Depot enthaltenen Titel regelmäßig.
 - Basierend auf bestimmten Kriterien (z. B. Kursentwicklungen, Volatilität, technische Indikatoren) trifft der Bot eine Kauf- oder Verkaufsentscheidung.
- **Kauf-/Verkaufsentscheidungen:**
 - Wird eine Kaufentscheidung getroffen, kauft der Bot den entsprechenden Titel.
 - Bei einer Verkaufsentscheidung verkauft der Bot den Titel und ermittelt dabei den Gewinn oder Verlust der Transaktion.

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

- **Speicherung der Handelsaktionen:**

- Jede getroffene Kauf- oder Verkaufsentscheidung wird in einer Historie-Tabelle pro Depot festgehalten.
- Zu den Informationen, die gespeichert werden, gehören:
 - Kauf-/Verkaufspreis
 - Gesamtwert der Position zum Kaufzeitpunkt
 - Gewinn/Verlust beim Verkauf

3.3 Benutzerverwaltung (optional)

- **Anmeldung und Authentifizierung:**

- Zugriff auf die Weboberfläche erfolgt nur nach erfolgreicher Anmeldung mit E-Mail und Passwort.
- Authentifizierte Benutzer haben Zugriff auf ihre Depots und Historie.

3.4 Historienübersicht

- **Verlauf der Handelsentscheidungen:**

- Benutzer können eine Historie aller Kauf- und Verkaufsentscheidungen pro Depot einsehen.
- Die Historie zeigt für jede Transaktion den Titel, den Zeitpunkt der Entscheidung, den Kauf- bzw. Verkaufspreis, den aktuellen Kurs und den Gewinn/Verlust.

4. Technische Anforderungen

4.1 Weboberfläche

- **Benutzerfreundliche Oberfläche:**

- Die Weboberfläche muss übersichtlich und einfach zu bedienen sein.
- Benutzer können Depots erstellen, Titel hinzufügen/entfernen, und die Historie einsehen.


- **Zugriffssicherheit:**

- Optional: Benutzer müssen sich mit E-Mail und Passwort authentifizieren, um Zugriff auf ihre Depots und Daten zu erhalten.

4.2 Datenbank

- **Datenbankmanagement:**

- Als Datenbank wird eine SQL-Datenbank verwendet.
- Die Datenbank speichert:
 - Benutzerinformationen (bei aktiviertem Login-System)
 - Depotinformationen (Depotname, enthaltene Titel)

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

- Transaktionshistorie (Kauf-/Verkaufsentscheidungen, Preise, Gewinne/Verluste)
- Kursdaten der handelbaren Titel (sofern benötigt).

4.3 Trading-Logik und Datenquellen

- **Datenquellen:**
 - Der Tradingbot analysiert alte Historische Markdaten, welche er von Bibliotheken wie Yahoofinance bezieht
- **Automatisierte Handelsentscheidungen:**
 - Die Logik zur Analyse und Entscheidungsfindung wird auf Basis von Algorithmen entwickelt, die historische und aktuelle Daten der Titel berücksichtigen.

5. Nicht-funktionale Anforderungen

5.2 Sicherheit

- Bei aktiviertem Login-System müssen Benutzeranmeldungen verschlüsselt erfolgen (z. B. mit SSL).

5.3 Skalierbarkeit

- Die Anwendung soll in der Lage sein, eine beliebige Anzahl von Depots und Transaktionen zu verwalten, ohne dass die Performance leidet.


6. Testanforderungen

- **Funktionale Tests:**
 - Test der Handelslogik: Bot trifft Kauf-/Verkaufsentscheidungen basierend auf den implementierten Algorithmen.
- **Sicherheitstests:**
 - Test der Authentifizierungsmechanismen (bei aktivierter Benutzeranmeldung).

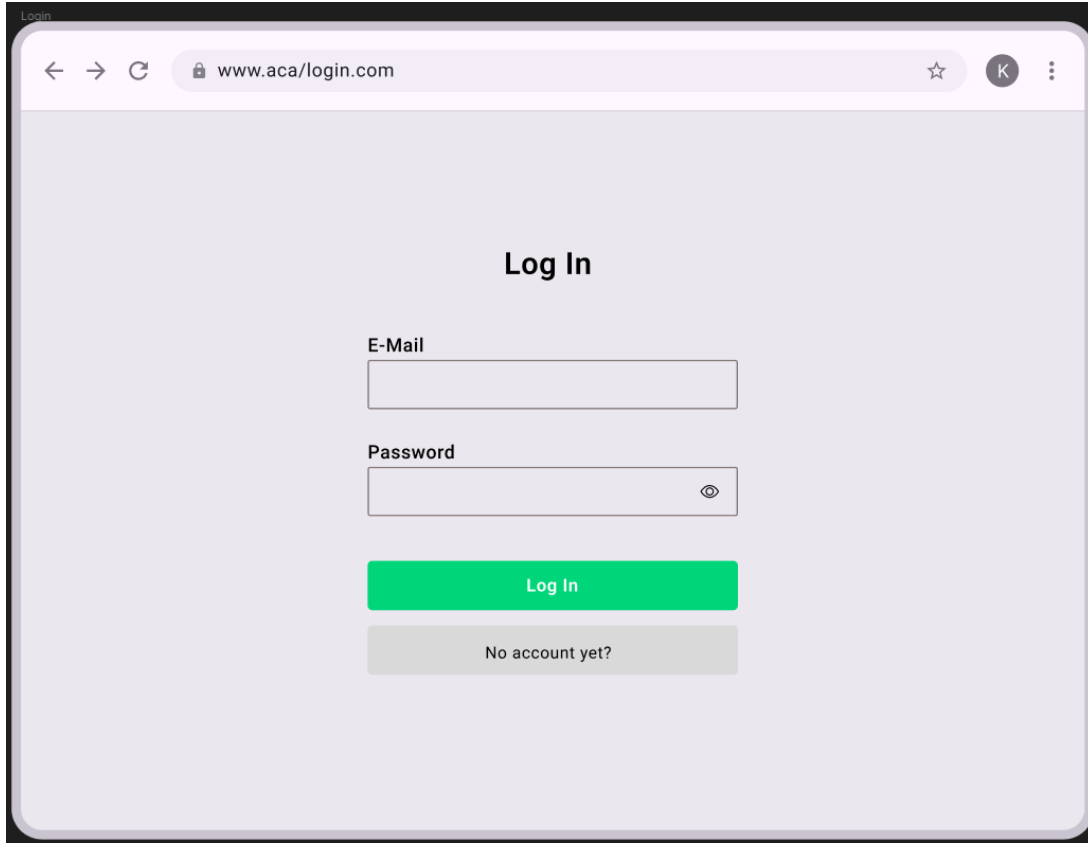
7. Dokumentation

- **Entwicklerdokumentation:**
 - Technische Dokumentation der Datenbankstruktur, APIs, und der Handelsalgorithmen wird erstellt.



	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

A.4. User Interface



Login

← → ↻ 🔒 www.aca/login.com ☆ K ⋮

Log In

E-Mail

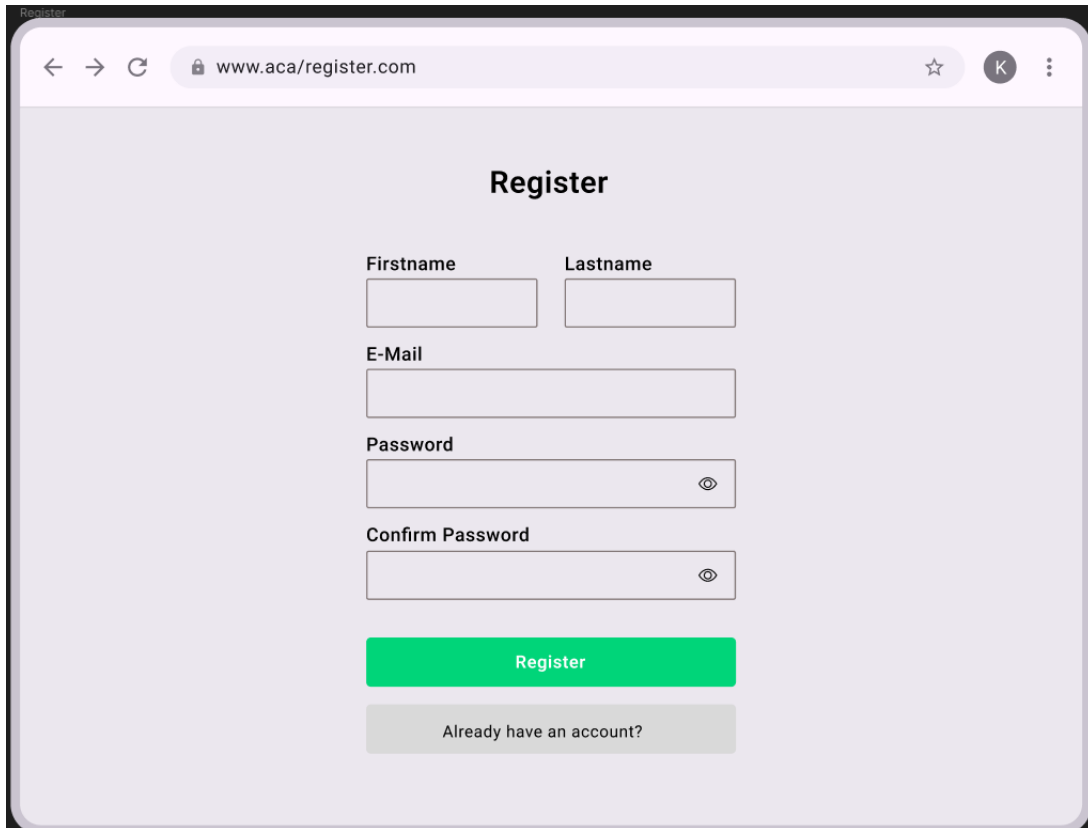
Password

Log In

No account yet?

Login:

- E-Mail-Feld
- Passwort-Feld
- Auge Button: Das eingegebene Passwort wird angezeigt
- Log In: Man wird auf das Dashboard weitergeleitet
- No account yet?: Man wird zur registrationsseite weitergeleitet

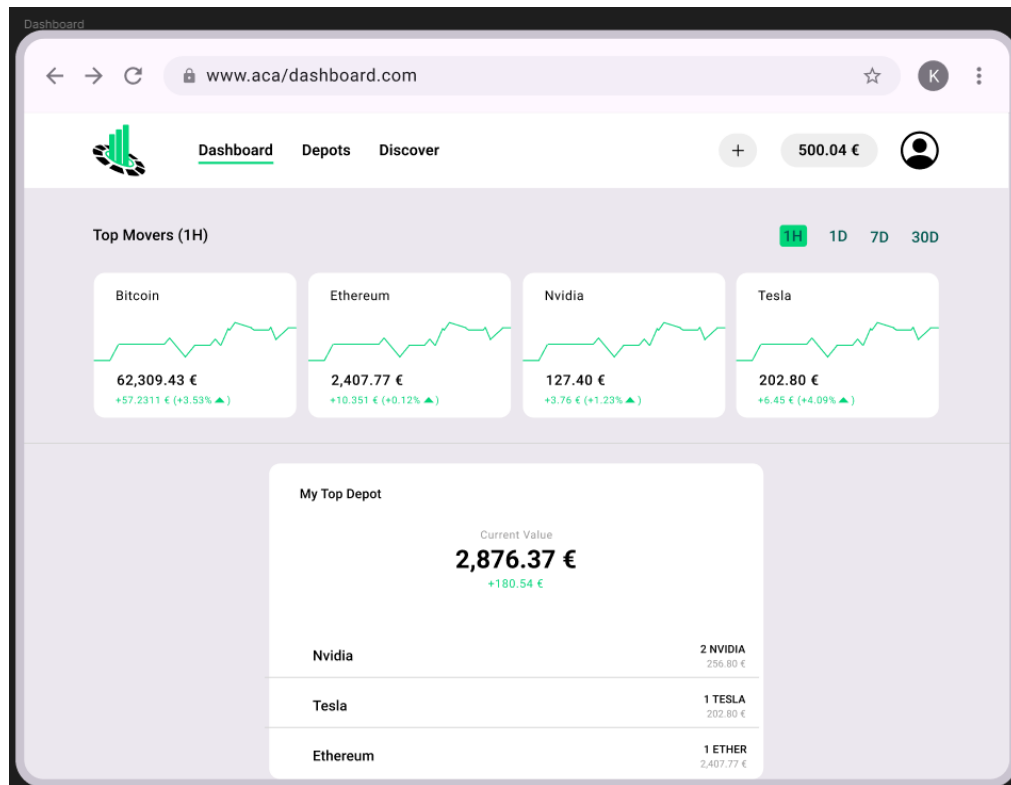


The screenshot shows a web browser window with the address bar displaying "www.aca/register.com". The page title is "Register". The form contains the following elements:

- Firstname**: A text input field.
- Lastname**: A text input field.
- E-Mail**: A text input field.
- Password**: A text input field with a toggle icon (eye) on the right.
- Confirm Password**: A text input field with a toggle icon (eye) on the right.
- Register**: A green button.
- Already have an account?**: A grey button.

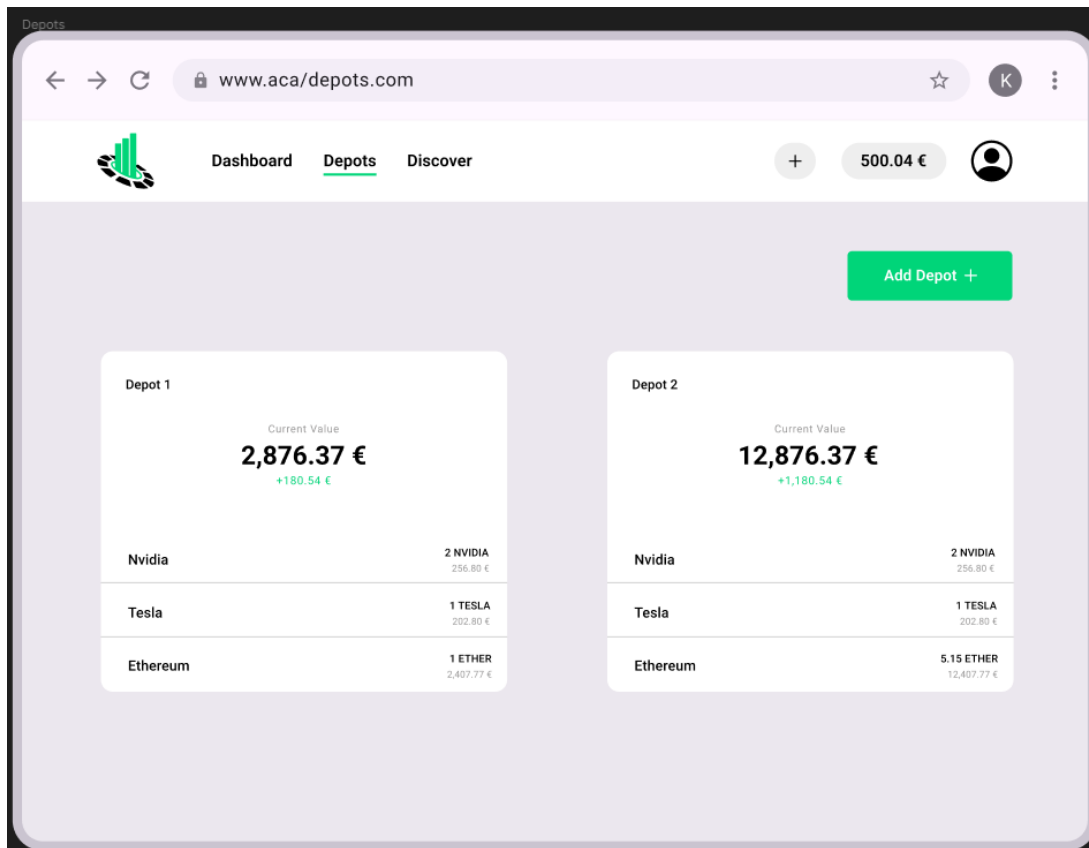
Register:

- Vorname-Feld
- Nachname-Feld
- E-Mail-Feld
- Passwort-Feld: Muss übereinstimmen -> wenn nicht, Alert („Passwort stimmt nicht überein“)
- Auge Button: Das eingegebene Passwort wird angezeigt
- Register Button: Man wird zur Login Page weitergeleitet
- Alredy have an account yet?: Man wird zur Login Page weitergeleitet



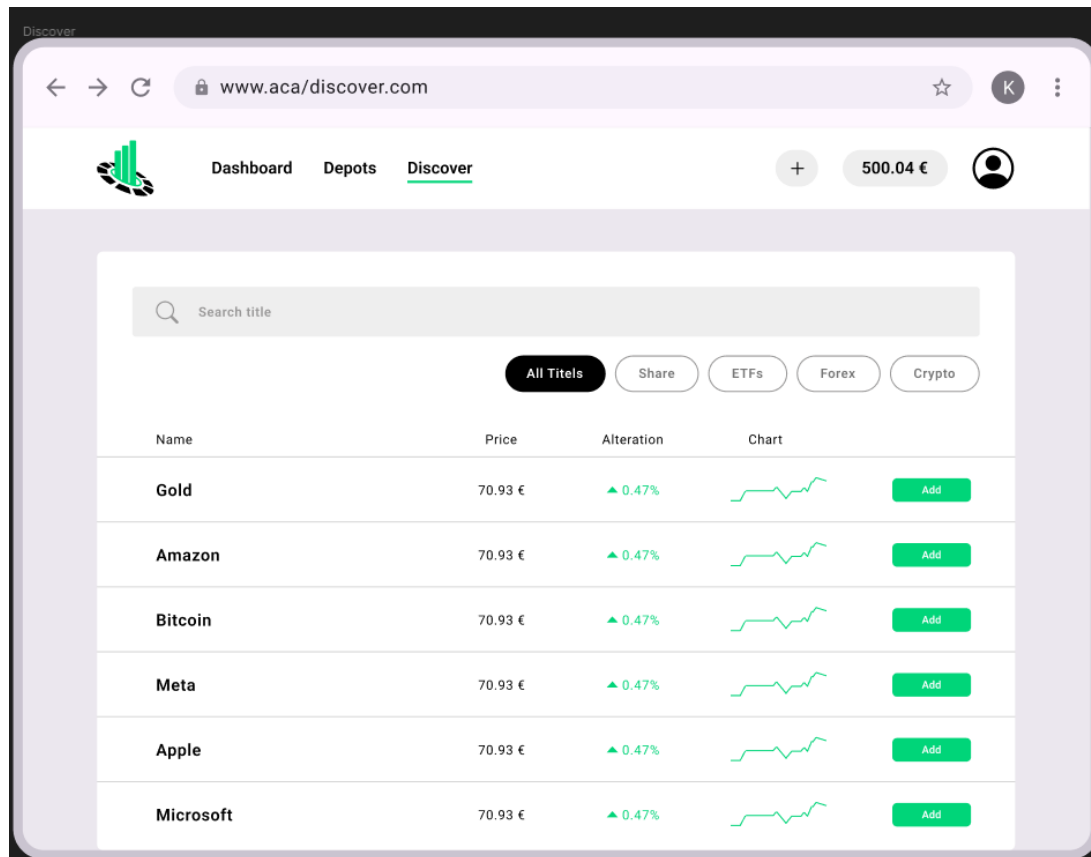
Dashboard:

- **Navbar**
 - **Dashboard:** Aktuelle Seite, auf der sich der Benutzer befindet.
 - **Depots:** Link zur Seite, auf der der Benutzer seine Depots einsehen und verwalten kann.
 - **Discover:** Link zu einer Seite, um neue Titel zu sehen
 - **Plus:** Bei Klick: Pop Up Fenster, in dem der Benutzer den gewünschten Betrag zur Einzahlung auf sein Konto eingeben kann. Das Fenster enthält ein Eingabefeld zur Eingabe des Betrags.
 - **Bestehender Geldbetrag:** rechts oben
 - **Profil:** Icon, das zur Profilsseite führt, wo der Benutzer seine Daten verwalten kann.
- **Top Movers:** Diese Sektion zeigt die wichtigsten, sich am stärksten bewegendende Vermögenswerte (Kryptowährungen, Aktien, etc.) im aktuellen Zeitrahmen (1 Tag).
 - **Bitcoin:** Preis von 62.309,43 €, grün markiert mit einem Anstieg von +57,2311 € (entspricht +3,53 %).
 - Vermögenswerte werden als kleine Karten dargestellt, die jeweils den Namen, den aktuellen Preis und die Preisänderung mit einem kleinen Diagramm des Kursverlaufs zeigen.
 - Daten werden unter dem Zeitrahmen "1H, 1D, 7D, 30D" angezeigt, mit "1H" als derzeit ausgewähltem Zeitrahmen
- **My Top Depot:**
 - **Depot-Wert:** Das beste Depot wird im mittleren Bereich angezeigt, mit einem **aktuellen Gesamtwert von 2.876,37 €** und einem Gewinn von **+180,54 €**.
 - **Einzelne Vermögenswerte** im Depot werden darunter aufgelistet



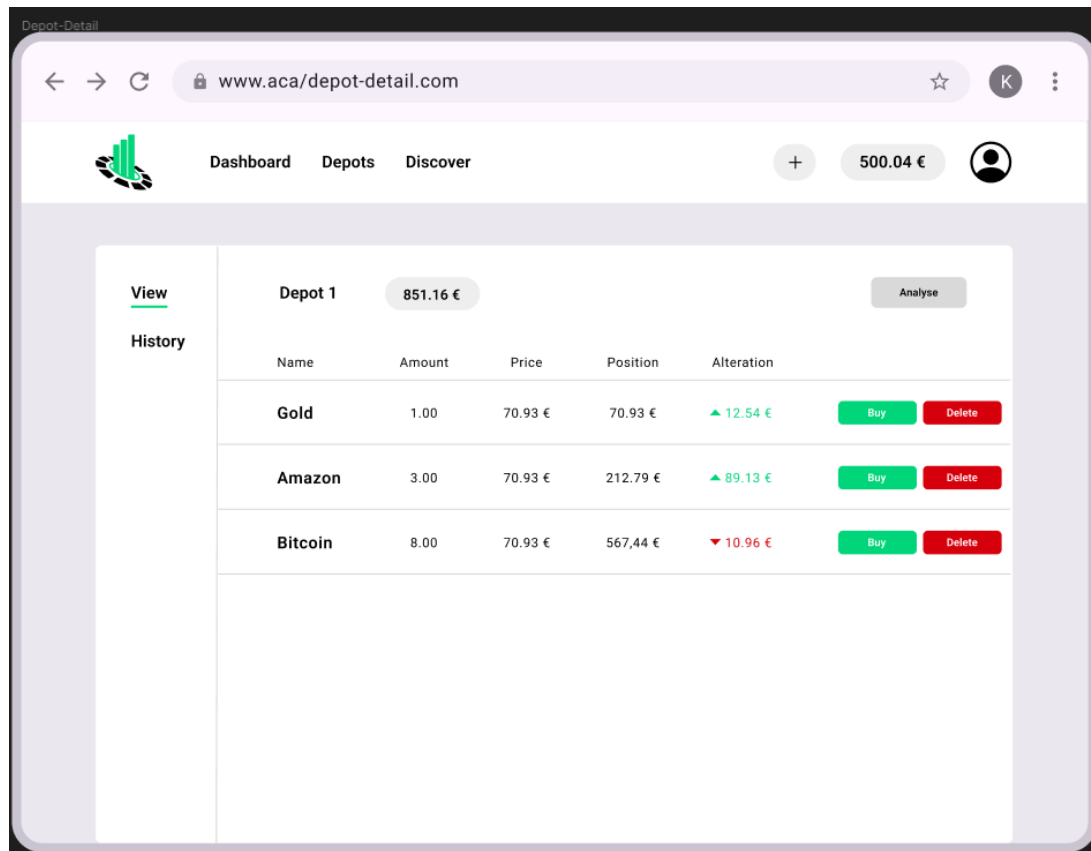
Depots:

- **Navbar**
- **Add Depot:** Bei Klick: Pop Up Fenster, um ein neues Depot hinzuzufügen. Eingabefeld für Name des Depots.
- **Depotübersicht:** Alle Depots werden gelistet. Die nächsten Depots werden darunter aufgelistet



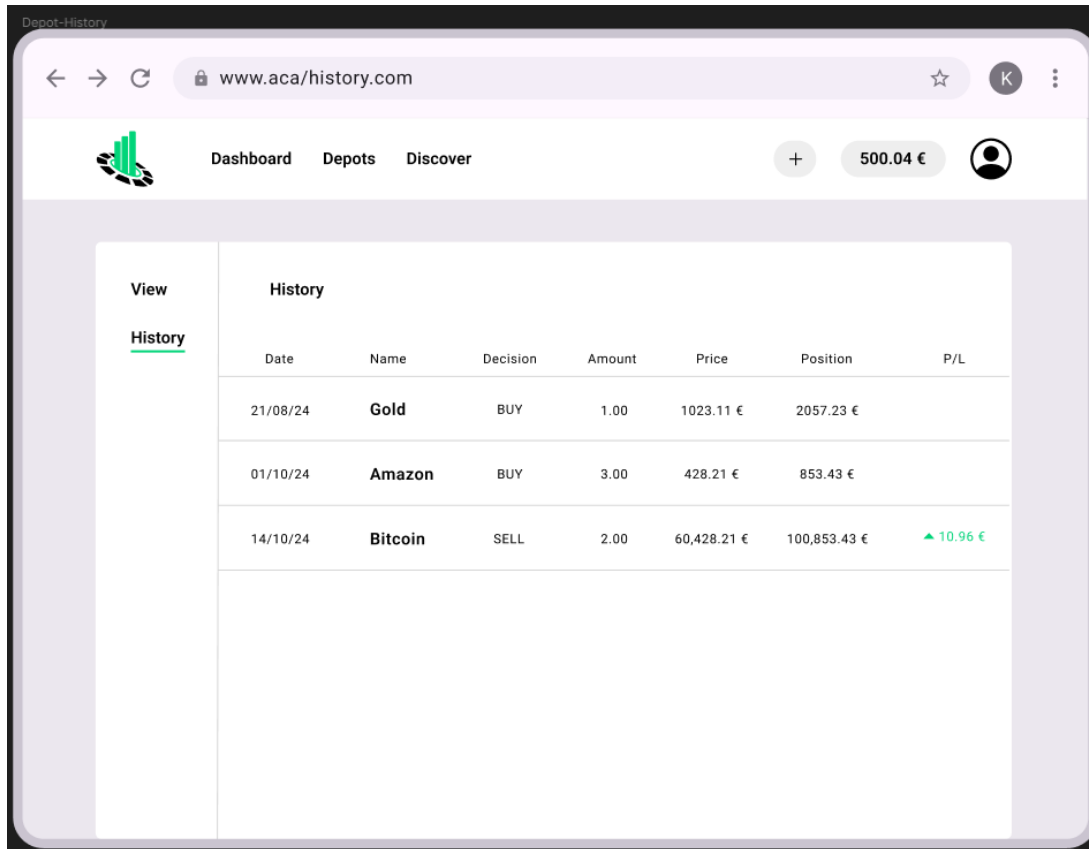
Discover:



- **Navbar**
- **Body:** Es werden alle in der Datenbank enthaltenen Titel angezeigt.
 - Search Title: nach dem Namen eines Titels suchen
 - Filterung der Titel nach: ETF, Share, Forex, Crypto
 - Liste aller gefundenen Titel:
 - Name: Name des Titels
 - Price: Aktueller Preis des Titels
 - Alteration: Änderung des Preises nach einer Woche
 - Chart: Überblick über den Kurs des Titels
 - Add: Bei Klick: Pop Up Fenster, um den Titel in ein bestehendes Depot hinzuzufügen.



Depot-Detail:

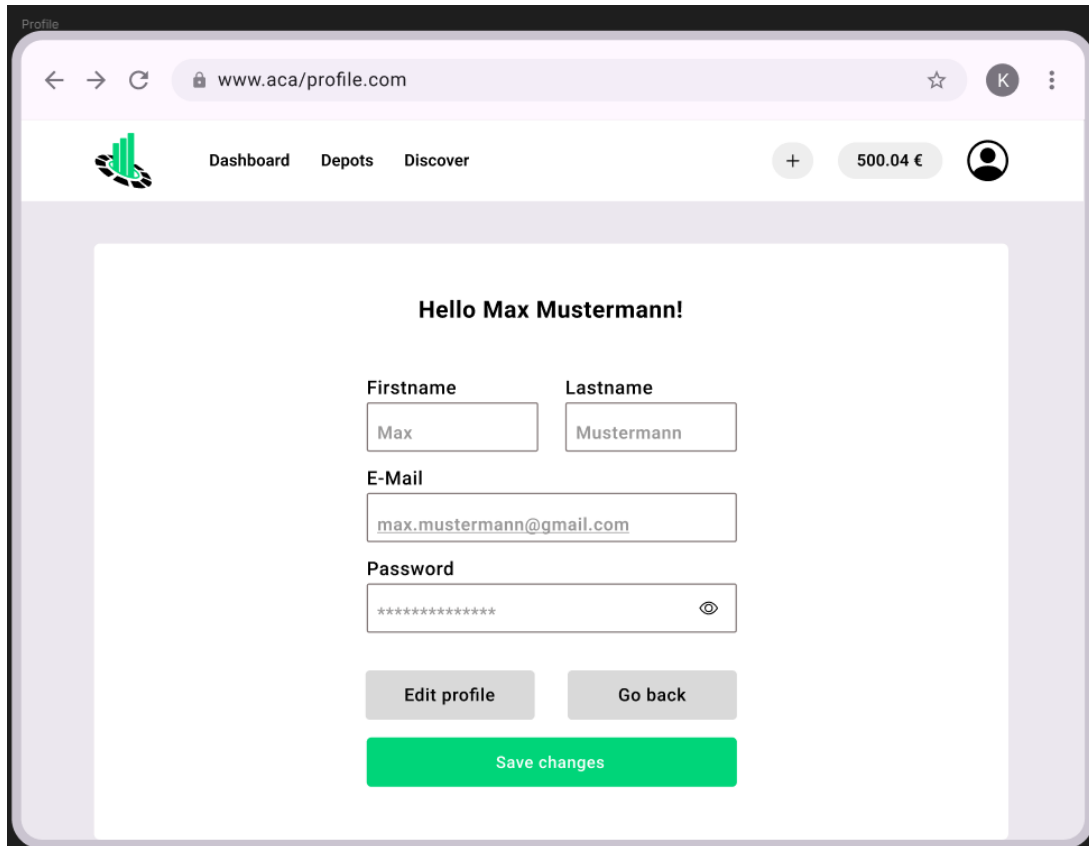
- **Navbar**
- **Sidebar:**
 - View: Aktuelle Seite
 - History: Verweis auf Depot-History Seite
- **Name des Depots**
- **Geldwert des Depots**
- **Analyse:** Bei Klick: Tradingbot analysiert, ob ein Kauf bzw. Verkauf eines Titels passend wäre.
 - **Kauf Entscheidung:** Titel Hintergrund wird grün eingefärbt
 - **Verkauf Entscheidung:** Titel Hintergrund wird rot eingefärbt
- **Liste der Titel im Depot:**
 - Name: Name des Titels
 - Amount: Menge der vorhandenen Titel
 - Price: Aktueller Preis des Titels
 - Position: Zusammengerechneter Wert der Titel. (Amount * Price)
 - Alteration: Vergleich zwischen dem Kaufpreis und dem aktuellen Preis
 - Buy: Bei Klick: Pop Up Fenster, um diesen Titel zu kaufen. Es kommt ein Feld, indem man die Anzahl der kaufenden Titel eingeben kann.
 - Delete: Entfernung des Titels aus dem Depot



Depot-History							
www.aca/history.com							
<div>  Dashboard Depots Discover + 500.04 €  </div>							
View	History						
History	Date	Name	Decision	Amount	Price	Position	P/L
	21/08/24	Gold	BUY	1.00	1023.11 €	2057.23 €	
	01/10/24	Amazon	BUY	3.00	428.21 €	853.43 €	
	14/10/24	Bitcoin	SELL	2.00	60,428.21 €	100,853.43 €	▲ 10.96 €

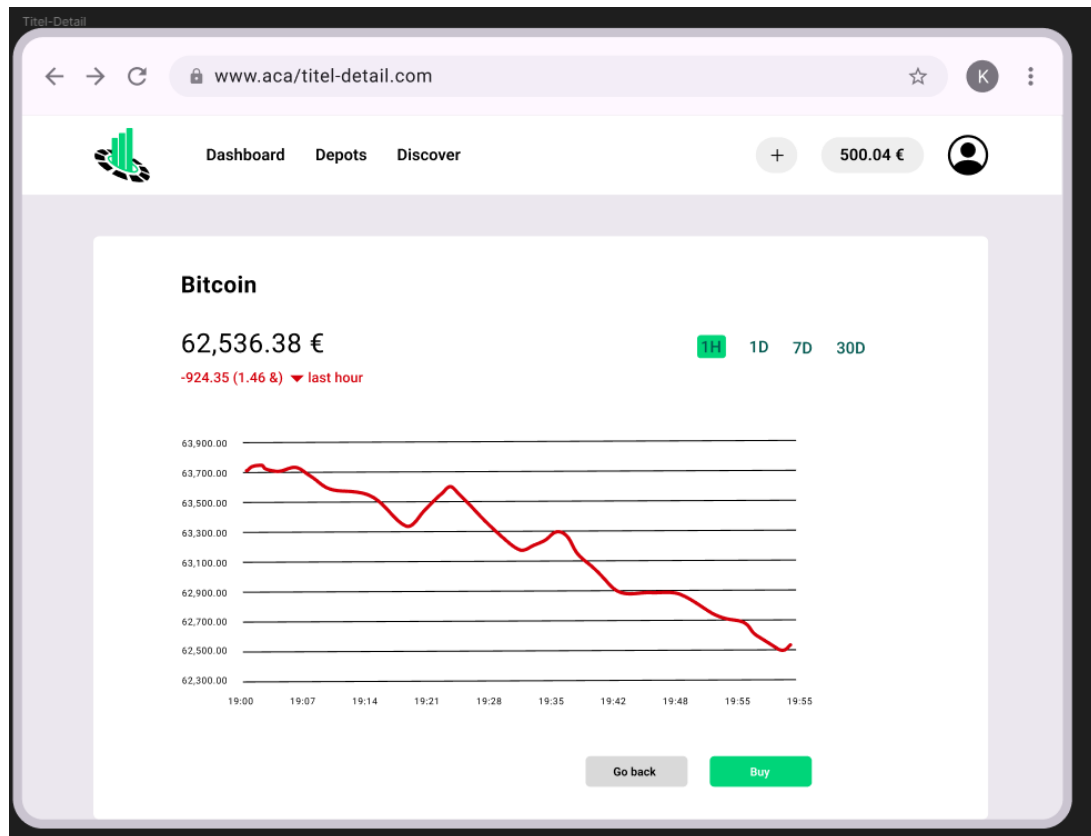
History:

- **Navbar**
- **Sidebar**
- **Verlaufstabelle:**
 - Datum: Das Datum, an dem die Transaktion stattgefunden hat.
 - Name: Name des Titels
 - Decision: Die Handelsentscheidung, ob der Vermögenswert **gekauft** ("BUY") oder **verkauft** ("SELL") wurde.
 - Amount: Die Anzahl der Anteile, die gekauft oder verkauft wurden
 - Price: Der Preis pro Einheit des Titels zum Zeitpunkt des Kaufs/Verkaufs
 - Position: Zusammengerechneter Wert der Titel. (Amount * Price)
 - **Gewinn/Verlust (P/L):** Zeigt den Gewinn oder Verlust für die entsprechende Transaktion an. Ein Gewinn wird durch grüne Markierung angezeigt, ein Verlust durch eine rote Markierung.




Profile:

- **Navbar**
- **Begrüßung:** „Hello Max Mustermann!“ – eine personalisierte Begrüßung des Benutzers mit seinem Namen.
- **Formulareingabefelder:**
 - Firstname: Eingabefeld für den Vornamen des Benutzers (aktuell „Max“).
 - Lastname: Eingabefeld für den Nachnamen des Benutzers (aktuell „Mustermann“).
 - E-Mail: Ein nicht bearbeitbares Feld, das die E-Mail-Adresse des Benutzers zeigt („max.mustermann@gmail.com“).
 - Password: Ein Passwortfeld, das durch ein verdecktes Eingabefeld und ein Symbol zum Anzeigen des Passworts gekennzeichnet ist.
- **Buttons:**
 - Edit profile: Ein grauer Button, der es dem Benutzer ermöglicht, sein Profil zu bearbeiten, wenn er ihn anklickt.
 - Go back: Ein zweiter Button, um zur vorherigen Seite zurückzukehren.
 - Save changes: Speichert die veränderten Eingabedaten.



Titel-detail:


- **Navbar**
- **Body**
 - **Titelname:** Der Name des Titels wird prominent angezeigt (hier „Bitcoin“), um klarzumachen, um welchen Titel es sich handelt.
 - **Aktueller Preis:** Der aktuelle Preis des Titels wird direkt unter dem Titelnamen angezeigt (62.536,38 €).
 - **Preisänderung:** Die Preisänderung in den letzten 1 Stunde wird in Rot angezeigt (-924,35 € bzw. -1,46 %), um den Verlust des letzten Zeitraums deutlich zu machen. Bei Gewinn in grüner Farbe.
 - **Zeitfenster-Optionen:** Der Benutzer kann zwischen verschiedenen Zeiträumen (1H, 1D, 7D, 30D) wechseln, um die Kursentwicklung des Titels zu analysieren. Aktuell ist das Zeitfenster „1H“ ausgewählt.
 - **Kurschart:** Ein Liniendiagramm, das den Kursverlauf des Titels innerhalb des ausgewählten Zeitrahmens darstellt. In diesem Fall zeigt es den Verlauf der letzten Stunde (1H) für Bitcoin.
- **Buttons:**
 - **Go back:** Dieser Button ermöglicht es dem Benutzer, zur vorherigen Seite zurückzukehren.
 - **Buy:** Bei Klick: Pop Up Fenster, um diesen Titel zu kaufen. Es kommt ein Feld, indem man die Anzahl der kaufenden Titel eingeben kann.

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

A.5. Datenträgerbeschreibung und Installationsanleitung

A.5.1. Datenträgerbeschreibung

A.5.2. Installationsanleitung

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

	HTL Krems Höhere Lehranstalt für Informationstechnologie Ausbildungsschwerpunkt	Reife- und Diplomprüfung
---	--	-------------------------------------

Installationsanleitung des Prototyps


Themenstellung:	ACA Tradingbot
Kandidaten/Kandidatinnen:	Karl Pferscher, Florian Brunner
Jahrgang:	5BHITM
Betreuer/in:	Panhofer Paul
Auftraggeber:	Dipl.-Ing. Mag. Georg Ungerböck X-WORKS GmbH
Ort:	HTL Krems

Voraussetzungen:

- Docker Desktop installiert
- Internetverbindung
- Container nicht parallel starten

Installationsschritte:

1. Ordner „TradingbotRoot“ im Terminal öffnen
2. Docker Desktop öffnen
3. Befehl in Terminal ausführen: „docker compose up --build“
4. In Docker Desktop:
 - a. Container „db“ starten
 - b. Container „blazor-app“ starten
 - c. Container „python-app“ starten
 - d. Container „python-flask“ starten

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:

A.6. Arbeitspaketbeschreibung und PSP

Informationstechnische Projekte

Höhere Technische Bundeslehranstalt Krems, 3500, Alauntalstraße 29A
mit Expositur Zwettl, 3910, Hammerweg 1

Projekt		Team
Nr	24/25-5BHITM	
Name	ACA - Trading Bot	
Dokument		
Nr	F_02	
Name	Arbeitspaketbeschreibung	

<i>Karl Pferscher</i> Florian Brunner
<i>Paul Pahv nhofer</i>


Versionsgeschichte

Version	Datum	Autor/in	Änderungen
0.1	28.09.2024	Karl Pferscher	F_02 erstellt


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
1	Projektplanung und Dokumentation	Karl Pferscher, Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
	1.1	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	720	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Projektplanung und Dokumentation		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
1.1	Erstellen des Pflichtenhefts	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
1	1.2	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	120	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Dokumentation aller Anforderungen an das System, einschließlich funktionaler und nicht-funktionaler Anforderungen. Das Pflichtenheft dient als vertragliche Grundlage zwischen Auftraggeber und Entwicklungsteam und beschreibt detailliert, was das System leisten soll.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
1.2	Erstellen des Projektstrukturplans	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
1.1	1.3	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	300	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Erstellen einer hierarchischen Darstellung des Projekts, das alle Haupt- und Unteraufgaben in Arbeitspakete unterteilt. Dies bietet eine Übersicht über die Struktur und den Ablauf des Projekts.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
1.3	Erstellen der Arbeitspakete und Meilensteine	Karl Pferscher
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
1.2	2	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	300	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Definieren der einzelnen Arbeitspakete mit ihren spezifischen Zielen, Verantwortlichkeiten und Zeitrahmen. Zudem werden Meilensteine gesetzt, um den Fortschritt zu überwachen und Zwischenziele zu erreichen.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

Planung


Abnahme

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
2	Datenbankdesign und Implementierung	Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
1.3	2.1	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	1740	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Datenbankdesign und Implementierung der Datenbank		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
2.1	Datenbankmodellierung	Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
2	2.2	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	420	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Analyse und Design der Datenbankstruktur, um alle erforderlichen Datenobjekte (Entitäten) und ihre Beziehungen zu definieren. Hierzu gehört auch die Erstellung eines Entity-Relationship-Diagramms (ERD).		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
2.1.1	Identifikation der Entitäten	Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
2.1	2.1.2	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	300	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Identifikation und Beschreibung der wichtigsten Datenobjekte (wie Benutzer, Depots, Transaktionen, Titel), die in der Datenbank gespeichert werden.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
2.1.2	Erstellung des ER-Diagramms	Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
2.1.2	2.2	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	360	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Visualisierung der Entitäten und ihrer Beziehungen zueinander in einem Entity-Relationship-Diagramm, das die Grundlage für die Datenbankstruktur bildet.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
2.2	Implementierung der Datenbank	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
2.1.2	2.2.1	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	600	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Technische Umsetzung der Datenbankstruktur. Dies umfasst das Anlegen der Tabellen, Beziehungen und Constraints (wie Primär- und Fremdschlüssel) in einem SQL-Datenbankmanagementsystem.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
2.2.1	Erstellung der Tabellen und Beziehungen	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
2.2	2.2.2	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	240	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Implementierung der Tabellen und deren Beziehungen gemäß dem erstellten ER-Diagramm. Definieren von Schlüsseln und Datentypen.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
2.2.2	Optimierung der Datenbank	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
2.2.1	2.3	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	360	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Datenbankoptimierung steigert die Leistung durch Indizierung, Query-Optimierung, Partitionierung und Caching. Dabei werden Daten auf korrekte Datentypen geprüft und, falls nötig, Tabellenstrukturen angepasst.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
2.3	Laden von historischen Marktdaten	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
2.2.2	2.3.1	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	720	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Laden und Validieren historischer Marktdaten (z.B. Aktienkurse, ETFs), um diese für die Handelslogik und Analysen zu verwenden.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
2.3.1	Auswahl der Datenquellen	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
2.3	2.3.2	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	180	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Auswahl geeigneter Datenquellen (wie Yahoo Finance) für die historischen Marktdaten.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
2.3.2	Extraktion und Transformation der Daten	Karl Pferscher
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
2.3.1	2.3.3	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	540	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Extrahieren und Anpassen der Daten in das für die SQL-Datenbank benötigte Format.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
2.3.3	Laden der Daten in die Datenbank	Karl Pferscher
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
2.3.2	3	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	600	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Laden der Daten in die SQL die SQL Datenbank und Nutzung der lokalen Parallelisierung des Apache Spark Frameworks.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
3	Web-Frontend Entwicklung	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
2.3.2	3.1	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	7140	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Web-Frontend Entwicklung (Mockups, Seiten Implementierung)			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
3.1	Mockups erstellen	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
3	3.2	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	1020	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Design von Prototypen der Benutzeroberfläche, um das Layout und die Benutzerführung zu visualisieren und abzustimmen.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
	1060		
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
3.1.1	Analyse der Benutzeranforderungen für das Frontend	Karl Pferscher
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
3.1	3.1.2	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	180	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Identifikation der Funktionen und Designanforderungen, um ein benutzerfreundliches und intuitives Frontend zu entwickeln.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
3.1.2	Erstellung und Optimierung der Mockups	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
3.1.1	3.1.3	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	840	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Design von Prototypen der Benutzeroberfläche und iterative Verbesserung basierend auf Feedback, um ein benutzerfreundliches und ansprechendes Layout zu gewährleisten.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
3.2	Auswahl des CSS-Frameworks	Karl Pferscher
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
3.1.2	3.3	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	120	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Auswahl eines CSS-Frameworks (z.B. MudBlazor), das zur Gestaltung der Benutzeroberfläche verwendet wird.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
3.3	Entwicklung des Frontends	Karl Pferscher
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
3.2	3.3.1	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	6000	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Technische Umsetzung der Benutzeroberfläche (Frontend) in .NET Blazor. Dies umfasst die Programmierung von Komponenten und das Styling der Anwendung.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
3.3.1	Implementierung der Registrierungsseite	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
3.3	3.3.2	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	460	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Entwicklung der Anmeldeseite, auf der Benutzer sich registrieren können.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
3.3.2	Implementierung der Login Seite	Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
3.3.1	3.3.3	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	460	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Entwicklung der Anmeldeseite, auf welcher Benutzer ihre Zugangsdaten eingeben und sich authentifizieren können, einschließlich der Integration von Sicherheitsmaßnahmen wie Passwortverschlüsselung.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
3.3.3	Implementierung der Depotübersicht	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
3.3.2	3.3.4	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	850	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Entwicklung der Seite, auf der Benutzer alle ihre Depots sehen, erstellen und bearbeiten können.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
3.3.4	Implementierung der Titelauflistung (Discover)	Karl Pferscher
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
3.3.3	3.3.5	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	850	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Die Implementierung der Titelauflistung umfasst das Abrufen aller verfügbaren Titel aus der Datenbank und deren Anzeige im Frontend, sodass Benutzer Titel für ihre Depots auswählen können.		
Tätigkeitsbeschreibung		
Entwicklung der Seite, die eine Liste von alle verfügbaren		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

Planung


Abnahme

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
3.3.5	Implementierung des Dashboards	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
3.3.4	3.3.6	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	850	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Die Implementierung der Dashboard-Seite ermöglicht es dem Benutzer, eine Übersicht über die wichtigsten Marktbewegungen (Top Movers) und den Wert seines Depots einzusehen. Dies umfasst die Anzeige von aktuellen Kursen und Preisänderungen in Echtzeit sowie die Darstellung des Depotwerts inklusive der einzelnen Positionen. Der Benutzer kann zwischen verschiedenen Zeiträumen (1H, 1D, 7D, 30D) wechseln, um die Kursentwicklungen visuell in Diagrammen zu verfolgen.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
3.3.6	Implementierung der Depot-Detail-Seite	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
3.3.5	3.3.7	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	540	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Die Implementierung der Depot-Detail-Seite ermöglicht es dem Benutzer, eine detaillierte Übersicht über die in einem Depot enthaltenen Titel zu erhalten. Auf dieser Seite kann der Benutzer die aktuellen Positionen einsehen, einschließlich Menge, Preis, Position und der Wertänderung (Alteration). Zusätzlich hat der Benutzer die Möglichkeit, einzelne Titel zu kaufen oder zu löschen. Der „Analyse“-Button ermöglicht es dem Benutzer, den Tradingbot eine Analyse durchführen zu lassen, um Empfehlungen zum Kauf oder Verkauf der jeweiligen Titel zu erhalten.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
3.3.7	Implementierung der Entscheidungshistorie	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
3.3.6	3.3.8	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	670	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Entwicklung einer Funktion und Anzeige aller Kauf- und Verkaufsentscheidungen. Dies umfasst die Darstellung von Datum, Entscheidungstyp (Kauf/Verkauf), Titel und relevanten Marktdaten.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
3.3.8	Implementierung der Profil-Seite	Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
3.3.7	3.3.9	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	650	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Die Implementierung der Profil-Seite ermöglicht es dem Benutzer, seine persönlichen Informationen wie Vorname, Nachname, E-Mail und Passwort zu verwalten. Der Benutzer kann seine Daten bearbeiten, indem er auf den Button „Edit profile“ klickt, oder zur vorherigen Seite zurückkehren, indem er den „Go back“-Button verwendet. Zusätzlich gibt es einen Button „Deposit money“, mit dem der Benutzer Geld auf sein Konto einzahlen kann.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
3.3.9	Implementierung der Titelanalyse-Seite	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
3.3.8	3.4	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	670	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Implementierung einer Titelanalyse Seite soll dem User einen Einblick in den gewünschten Titel geben. Es werden der Linienschart sowie der Aktuelle Preis und Kursänderungen angezeigt.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
4	Tradinglogik in Apache Spark	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
3.4.7	4.1	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	3600	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Identifikation der Handelslogik, technische Indikatoren und Kursanalysen mithilfe von Apache Spark			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
4.1	Definition der Handelslogik	Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
4	4.1.1	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	1200	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Festlegung der Regeln und Parameter, die der Tradingbot für Kauf- und Verkaufsentscheidungen basierend auf technischen Indikatoren verwendet.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
4.1.1	Auswahl der technischen Indikatoren	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
4.1	4.1.2	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	420	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Auswahl und Implementierung der technischen Indikatoren (wie RSI, MACD), die zur Analyse der Marktdaten verwendet werden.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
4.1.2	Berechnung der Indikatoren	Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
4.1.1	4.1.3	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	240	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Implementierung der Berechnung technischer Indikatoren wie RSI, MACD und gleitende Durchschnitte, die zur Analyse von Marktdaten verwendet werden und Kauf-/Verkaufsentscheidungen unterstützen.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
4.1.3	Erstellung der Handelsregeln für Kauf/Verkauf	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
4.1.2	4.2	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	540	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Definition der Logik und Regeln, die der Bot verwenden wird, um zu entscheiden, wann und welche Titel gekauft oder verkauft werden sollen.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
4.2	Implementierung der Tradinglogik	Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
4.1.3	4.2.1	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	1200	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Entwicklung und Implementierung der Handelslogik mit Apache Spark, um Marktdaten zu analysieren und Entscheidungen zu treffen.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
4.2.1	Implementierung der Kursanalysen	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
4.2	4.2.2	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	600	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Entwicklung der Algorithmen, die die Kursbewegungen analysieren und Kauf-/Verkaufsentscheidungen vorbereiten.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
4.2.2	Lokale Parallelisierung der Berechnungen	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
4.2.1	4.2.3	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	360	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Optimierung der Spark-Logik durch Parallelisierung, um große Datenmengen effizient zu verarbeiten.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
4.2.3	Performanceoptimierung	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
4.2.2	4.3	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	240	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Optimierung der Tradinglogik hinsichtlich der Ausführungsgeschwindigkeit und der Ressourcennutzung.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
4.3	Backtesting der Handelslogik	Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
4.2.3	4.3.1	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	1200	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Testen der Handelslogik anhand historischer Daten, um die Effektivität und Robustheit der Handelsstrategien zu überprüfen.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
4.3.1	Implementierung des Backtesting-Moduls	Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
4.3	4.3.2	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	300	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Entwicklung eines Backtesting-Moduls, das die Handelslogik auf historische Marktdaten anwendet und die Ergebnisse protokolliert.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
4.3.2	Durchführung der Backtests	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
4.3.1	4.3.3	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	600	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Ausführung der Backtests mit historischen Marktdaten, um die Handelslogik zu validieren.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
4.3.3	Ergebnisanalyse und Optimierung	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
4.3.2	5	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	300	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Analyse der Backtestergebnisse und Optimierung der Handelslogik basierend auf den gewonnenen Erkenntnissen.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
5	Backend und API-Entwicklung	Karl Pferscher
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
4.3.3	5.1	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	4000	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Entwicklung der Backend Logik und API.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
5.1	Entwicklung der Backendlogik	Karl Pferscher
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
5	5.1.1	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	1560	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Entwicklung der serverseitigen Logik zur Verarbeitung von Benutzeranfragen, Datenabfragen und Interaktionen mit der Datenbank.		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
5.1.1	Festlegung der Backendfunktionalitäten	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
5.1	5.1.2	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	480	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Festlegung der zentralen Backend-Funktionalitäten für die Blazor-Anwendung, einschließlich der Definition von der Datenbank-Schnittstelle zur Kommunikation zwischen Frontend und Backend, Authentifizierungs- und Datenverwaltungsfunktionen.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
5.1.2	Implementierung des Backends	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
5.1.1	5.2	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	1080	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Implementierung des Backends der Blazor-Anwendung.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
5.2	Entwicklung der API-Schnittstellen	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
5.1.2	5.2.1	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	640	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Implementierung von API-Endpunkten, die den Datenaustausch zwischen Frontend und Backend.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
5.2.1	Implementierung der API für Marktdaten	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
5.2	5.2.2	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	320	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
API-Endpoint zur Bereitstellung von Marktdaten für das Frontend und die Handelslogik.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
5.2.2	Response Testing und Fehlerbehebung	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
5.2.1	6	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	320	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Testen der API-Responses und Behebung von Fehlern, um sicherzustellen, dass die API zuverlässig funktioniert.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
6	Testen und Qualitätssicherung	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
5.2.2	6.1	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
		€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Funktionale Tests und Qualitätssicherung			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
6.1	Funktionale Tests	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
6	6.1.1	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	600	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Testen der Kernfunktionalitäten des Systems, einschließlich der Handelslogik, der Weboberfläche und der API, um sicherzustellen, dass alle Funktionen wie vorgesehen arbeiten.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
6.1.1	Testen der Handelslogik	Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
6.1	6.2	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	600	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Überprüfung der Implementierung der Handelslogik, um sicherzustellen, dass Kauf- und Verkaufsentscheidungen korrekt getroffen werden.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	


Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
6.2	Sicherheitstests	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
6.1.1	6.2.1	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	800	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Überprüfung der Sicherheit des Systems, insbesondere des Zugriffs auf Benutzerdaten und API, sowie der Sicherstellung der Integrität von Transaktionen und persönlichen Informationen.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
6.2.1	Testen der Datenintegrität	Karl Pferscher	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
6.2	7	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	200	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Sicherstellen, dass alle in die Datenbank eingegebenen Daten korrekt und vollständig sind und dass keine Daten während der Verarbeitung verloren gehen.			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung		
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich
7	Recherche und Dokumentation	Karl Pferscher Florian Brunner
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)
6.2.1	7.1	Karl Pferscher Florian Brunner
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten
	2400	€
Ziele, Ergebnisse		
Voraussetzungen für den Start des Arbeitspaketes		
Tätigkeitsbeschreibung		
Recherche über die Frameworks und Dokumentation der Arbeit		
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)
04.04.2025		

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
7.1	Erstellung der Entwickler Dokumentation	Karl Pferscher / Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
7	/	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	1800	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Erstellung einer genauen Entwickler Dokumentation und Beantwortung der Forschungsfragen			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

	HÖHERE TECHNISCHE BUNDES - LEHRANSTALT Krems	
	Abteilung: Informationstechnologie Ausbildungsschwerpunkt:	

Arbeitspaketbeschreibung			Planung
AP Nr (PSP-Code)	AP Bezeichnung (aus PSP)	Verantwortlich	
7.2	Recherche	Karl Pferscher / Florian Brunner	
Vorlieger (PSP-Code)	Nachfolger (PSP-Code)	Mitarbeit (gegebenenfalls)	
7.1	/	Karl Pferscher Florian Brunner	
Geplanter Starttermin	Geplante Dauer (in Minuten)	Geplante Kosten	
	600	€	
Ziele, Ergebnisse			
Voraussetzungen für den Start des Arbeitspaketes			
Tätigkeitsbeschreibung			
Recherche über Frameworks und verwendete Technologien			
Tatsächlicher Starttermin	Tatsächliche Dauer (in Minuten)	Tatsächliche Kosten	Abnahme
Abnahmetermin	Abgabe durch (Unterschrift)	Abnahme durch (Unterschrift)	
04.04.2025			

ACATRADINGBOT

Planung

Entwicklung

Dokumentation und Testen

Projektplanung

Datenbankdesign und Implementierung

Web-Frontend Entwicklung

Tradinglogik in ApacheSprak

Backend und API-Entwicklung

Testen und Qualitätssicherung

Erstellen des Pflichtenhefts
Verantwortlich: Florian Brunner
Id: 1.1
Arbeitszeit: 4h

Erstellen des PSPs (Projektstrukturplan)
Verantwortlich: Florian Brunner
Id: 1.2
Arbeitszeit: 8h

Erstellen der Arbeitspakete und Meilensteine
Verantwortlich: Karl Plescher
Id: 1.3
Arbeitszeit: 8h

Datenbankmodellierung
Verantwortlich: Florian Brunner
Id: 2.1
Arbeitszeit: 8h

Identifikation der Meilensteine
Verantwortlich: Florian Brunner
Id: 2.1.1
Arbeitszeit: 5h

Erstellen eines ER-Modells
Verantwortlich: Florian Brunner
Id: 2.1.2
Arbeitszeit: 2h

Implementierung der Datenbank
Verantwortlich: Florian Brunner
Id: 2.2
Arbeitszeit: 10h

Erstellung der Tabellen und Beziehungen
Verantwortlich: Florian Brunner
Id: 2.2.1
Arbeitszeit: 4h

Optimierung der Datenbank
Verantwortlich: Florian Brunner
Id: 2.2.2
Arbeitszeit: 6h

Laden von historischen Marktdaten
Verantwortlich: Karl Plescher
Id: 2.3
Arbeitszeit: 20h

Auswahl der Datenquellen
Verantwortlich: Karl Plescher
Id: 2.3.1
Arbeitszeit: 2h

Extraktion und Transformieren der Daten
Verantwortlich: Karl Plescher
Id: 2.3.2
Arbeitszeit: 5h

Laden der Daten in die Datenbank
Verantwortlich: Karl Plescher
Id: 2.3.3
Arbeitszeit: 10h

Mockups
Verantwortlich: Karl Plescher
Id: 3.1
Arbeitszeit: 10h

Analyse der Benutzeranforderungen für das Frontend
Verantwortlich: Karl Plescher
Id: 3.1.1
Arbeitszeit: 5h

Erstellung und Optimierung der Mockups
Verantwortlich: Karl Plescher
Id: 3.1.2
Arbeitszeit: 14h

Auswahl des CSS Frameworks
Verantwortlich: Karl Plescher
Id: 3.2
Arbeitszeit: 2h

Entwicklung des Frontends
Verantwortlich: Florian Brunner
Id: 3.3
Arbeitszeit: 87h

Implementierung der Registrierungsseite
Verantwortlich: Karl Plescher
Id: 3.3.1
Arbeitszeit: 6h

Implementierung der Loginseite
Verantwortlich: Karl Plescher
Id: 3.3.2
Arbeitszeit: 6h

Implementierung der Depotübersicht
Verantwortlich: Florian Brunner
Id: 3.3.3
Arbeitszeit: 14h

Implementierung der Teilaufstellung
Verantwortlich: Karl Plescher
Id: 3.3.4
Arbeitszeit: 14h

Implementierung des Dashboard
Verantwortlich: Karl Plescher
Id: 3.3.5
Arbeitszeit: 12h

Implementierung der Depot-Detail-Seite
Verantwortlich: Karl Plescher
Id: 3.3.6
Arbeitszeit: 9h

Implementierung der Entscheidungshistorie
Verantwortlich: Karl Plescher
Id: 3.3.7
Arbeitszeit: 11h

Implementierung der Profil-Seite
Verantwortlich: Florian Brunner
Id: 3.3.8
Arbeitszeit: 10h

Implementierung der Teilaufstellung
Verantwortlich: Karl Plescher
Id: 3.3.9
Arbeitszeit: 12h

Definition der Handelslogik
Verantwortlich: Florian Brunner
Id: 4.1
Arbeitszeit: 12h

Auswahl der Technischen Indikatoren
Verantwortlich: Florian Brunner
Id: 4.1.1
Arbeitszeit: 10h

Berechnung der Indikatoren
Verantwortlich: Florian Brunner
Id: 4.1.2
Arbeitszeit: 7h

Erstellung der Handelsregeln für Kauf/Verkauf
Verantwortlich: Florian Brunner
Id: 4.1.3
Arbeitszeit: 15h

Implementierung der Tradinglogik
Verantwortlich: Florian Brunner
Id: 4.2
Arbeitszeit: 11h

Implementierung der Kurvenanalysen
Verantwortlich: Florian Brunner
Id: 4.2.1
Arbeitszeit: 15h

Lokale Parallelisierung der Berechnungen
Verantwortlich: Florian Brunner
Id: 4.2.2
Arbeitszeit: 10h

Performanceoptimierung
Verantwortlich: Florian Brunner
Id: 4.2.3
Arbeitszeit: 6h

Backtesting der Handelslogik
Verantwortlich: Florian Brunner
Id: 4.3
Arbeitszeit: 23h

Implementierung des Backtestingmoduls
Verantwortlich: Florian Brunner
Id: 4.3.1
Arbeitszeit: 6h

Durchführung der Backtests
Verantwortlich: Florian Brunner
Id: 4.3.2
Arbeitszeit: 12h

Ergebnisanalyse und Optimierung
Verantwortlich: Florian Brunner
Id: 4.3.3
Arbeitszeit: 5h

Entwicklung der Backendlogik
Verantwortlich: Karl Plescher
Id: 5.1
Arbeitszeit: 20h

Festlegung der Backendfunktionalitäten
Verantwortlich: Karl Plescher
Id: 5.1.1
Arbeitszeit: 8h

Implementierung des Backends
Verantwortlich: Karl Plescher
Id: 5.1.2
Arbeitszeit: 18h

Entwicklung der API-Schnittstellen
Verantwortlich: Karl Plescher
Id: 5.2
Arbeitszeit: 12h

Implementierung der API für Marktdaten
Verantwortlich: Karl Plescher
Id: 5.2.1
Arbeitszeit: 6h

Response Testing und Fehlerbehebung
Verantwortlich: Karl Plescher
Id: 5.2.2
Arbeitszeit: 8h

Funktionale Tests
Verantwortlich: Florian Brunner
Id: 6.1
Arbeitszeit: 20h

Testen der Handelslogik
Verantwortlich: Florian Brunner
Id: 6.1.1
Arbeitszeit: 10h

Testen der Programmlogik
Verantwortlich: Florian Brunner
Id: 6.1.2
Arbeitszeit: 10h

Sicherheitstests
Verantwortlich: Karl Plescher
Id: 6.2
Arbeitszeit: 10h

Testen der Datenintegrität
Verantwortlich: Karl Plescher
Id: 6.2.1
Arbeitszeit: 10h