



Σχολή Θετικών Επιστημών & Τεχνολογίας
Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα
(MSc)

Διπλωματική Εργασία:

Δημιουργία εφαρμογής διαδικτύου
για την αξιολόγηση και ταξινόμηση μετοχών

Λάμπρος Γατσώρης

Επιβλέπων καθηγητής: Γεώργιος Ρηγόπουλος

Αθήνα, Σεπτέμβριος 2026

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή («συγγραφέας/δημιουργός») που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο ΕΑΠ, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.



Δημιουργία εφαρμογής διαδικτύου
για την αξιολόγηση και ταξινόμηση μετοχών

Λάμπρος Γατσώρης

Επιτροπή Επίβλεψης Πτυχιακής / Διπλωματικής Εργασίας

Επιβλέπων Καθηγητής:

Γεώργιος Ρηγόπουλος

Μέλος ΣΕΠ ΕΑΠ

Συν-Επιβλέπων Καθηγητής:

Βασίλειος Καψάλης

Μέλος ΣΕΠ ΕΑΠ

Αθήνα, Σεπτέμβριος 2026

Στις κόρες μου

Περίληψη

Η παρούσα διπλωματική εργασία έχει ως αντικείμενο τη δημιουργία διαδικτυακής εφαρμογής για την αξιολόγηση και ταξινόμηση μετοχών.

Στο θεωρητικό μέρος της εργασίας παρουσιάζονται βασικές έννοιες της Θεμελιώδους και της Τεχνικής Ανάλυσης, που είναι δύο συμπληρωματικές προσεγγίσεις για την ανάλυση και αξιολόγηση μετοχών, περιγράφονται οι τεχνικοί δείκτες και οι στρατηγικές συναλλαγών, που προσφέρονται για αυτοματοποίηση και έλεγχο πάνω σε ιστορικά δεδομένα, και οριοθετείται το εύρος της εφαρμογής.

Δεδομένου ότι πολλές από τις διαθέσιμες πλατφόρμες παρέχουν αρκετά ισχυρά εργαλεία ανάλυσης, η έμφαση δόθηκε στην επεκτασιμότητα και τη δυνατότητα πειραματισμού στην ανάλυση μετοχών χρησιμοποιώντας δημόσια διαθέσιμες τεχνολογίες, σε ένα πιο ανοιχτό οικοσύστημα όπως αυτό της Python. Η εφαρμογή, που έχει βασισθεί στο Streamlit framework, ανακτά χρηματιστηριακά δεδομένα μέσω της υπηρεσίας Yahoo Finance για το Χρηματιστήριο Αθηνών (ASE), τα αμερικανικά χρηματιστήρια NYSE και NASDAQ, καθώς και το χρηματιστήριο του Λονδίνου (LSE), και παρέχει λειτουργικότητα αναζήτησης μετοχών με βάση θεμελιώδη και τεχνικά μεγέθη, υπολογισμό τεχνικών δεικτών και δημιουργία γραφημάτων, δοκιμή στρατηγικών συναλλαγών πάνω σε ιστορικά δεδομένα, και ταξινόμηση των μετοχών με διάφορα κριτήρια. Στην εργασία τεκμηριώνονται οι κύριες τεχνολογικές επιλογές και αναπτύσσονται οι σχεδιαστικές αρχές και τα πρότυπα που χρησιμοποιήθηκαν, μεταξύ άλλων για να επιτευχθεί ο στόχος της επεκτασιμότητας.

Το βασικό συμπέρασμα από την αξιολόγηση της εφαρμογής είναι ότι αυτές οι επιλογές και οι σχεδιαστικές αρχές αποτελούν μια έγκυρη προσέγγιση στην ανάπτυξη εφαρμογών αυτού του είδους, και ότι ο κεντρικός στόχος της επεκτασιμότητας επιτεύχθηκε σε ικανοποιητικό βαθμό, αφού η υλοποίηση νέων δεικτών και στρατηγικών συναλλαγών απαιτεί ελάχιστο και καλά οριοθετημένο κώδικα. Παράλληλα, εντοπίζονται και πιθανές μελλοντικές βελτιώσεις.

Καθώς ο κώδικας της εφαρμογής είναι διαθέσιμος σε δημόσιο αποθετήριο, είναι σκόπιμο να τονιστεί ότι η εφαρμογή έχει αποκλειστικά ερευνητικό και πειραματικό χαρακτήρα και δεν προορίζεται για λήψη πραγματικών επενδυτικών αποφάσεων.

Λέξεις – Κλειδιά

Python, Streamlit, Yahoo Finance, μετοχές, τεχνική ανάλυση, θεμελιώδης ανάλυση, στρατηγικές συναλλαγών

Implementation of a Web Application for evaluating and classifying stocks

Lampros Gatsoris

Abstract

This thesis concerns the development of a web application for the evaluation and classification of stocks.

The theoretical part of the thesis presents basic concepts of Fundamental and Technical Analysis, which are two complementary approaches to stock analysis and evaluation. It also describes technical indicators and trading strategies, which can be automated and back tested on historical data, and outlines the scope of the application.

Given that many available platforms already provide quite powerful analysis tools, the emphasis was placed on extensibility and on the ability to experiment with stock analysis using publicly available technologies within a more open ecosystem, such as that of Python. The application, which is based on the Streamlit framework, retrieves stock market data through the Yahoo Finance service for the Athens Stock Exchange (ASE), the US stock exchanges NYSE and NASDAQ, as well as the London Stock Exchange (LSE). It provides functionality for searching stocks based on fundamental and technical metrics, calculating technical indicators and generating charts, testing strategies on historical data, and classifying stocks according to various criteria. The thesis provides justification for the main technological choices and presents the design principles and patterns that were used to achieve, among others, the objective of extensibility.

The main conclusion from the evaluation of the application is that these choices and design principles constitute a valid approach to the development of applications of this kind, and that the central objective of extensibility has been achieved to a satisfactory level, as the implementation of new indicators and trading strategies requires only minimal and well-scoped code. At the same time, potential future improvements are also identified.

Given that the source code of the application is available in a public repository, it must be emphasised that the application has an exclusively research-oriented and experimental purpose and is not intended to be used for making actual investment decisions.

Keywords

Python, Streamlit, Yahoo Finance, stocks, technical analysis, fundamental analysis, trading strategies

Πίνακας περιεχομένων

Περίληψη.....	v
Abstract	vi
Πίνακας περιεχομένων	vii
Κατάλογος Εικόνων / Σχημάτων	ix
Κατάλογος Πινάκων	x
Συντομογραφίες & Ακρωνύμια.....	xi
1. Εισαγωγή.....	1
2. Θεωρητικό υπόβαθρο.....	4
2.1. Μετοχές και χρηματιστηριακά δεδομένα.....	4
2.2. Στοιχεία θεμελιώδους ανάλυσης.....	5
2.3. Εισαγωγή στην τεχνική ανάλυση.....	8
2.4. Δείκτες τεχνικής ανάλυσης	12
2.4.1. Κινητός μέσος όρος.....	13
2.4.2. Εκθετικός κινητός μέσος όρος	14
2.4.3. Δείκτης ορμής (Momentum)	15
2.4.4. Δείκτης σχετικής ισχύος (RSI).....	16
2.4.5. Δείκτης σύγκλισης – απόκλισης κινητών μέσων (MACD)	17
2.5. Σήματα και στρατηγικές συναλλαγών	19
2.6. Επισημάνσεις	20
3. Αρχιτεκτονική και τεχνολογίες εφαρμογής	21
3.1. Python	21
3.1.1. Γενικά για την Python	21
3.1.2. Ανάλυση και επεξεργασία δεδομένων	23
3.1.3. Αντικειμενοστραφής σχεδίαση με την Python.....	23
3.1.4. Βιβλιοθήκες pandas και Altair	24
3.2. Διαδικτυακές εφαρμογές – Ιστορική αναδρομή	24
3.2.1. Server-side rendering και MVC frameworks.....	25
3.2.2. Ajax και μονοσελιδικές εφαρμογές (SPA).....	25
3.2.3. Εναλλακτικές προσεγγίσεις και UI-As-Code.....	26
3.3. Το Streamlit framework	28
3.3.1. Γενικά για το Streamlit.....	28
3.3.2. Αρχιτεκτονική και μοντέλο εκτέλεσης	28
3.3.3. Πλεονεκτήματα και περιορισμοί.....	31
3.3.4. Χρήση του Streamlit στην εφαρμογή.....	32
3.4. Yahoo Finance	33
3.4.1. Επιλογή της υπηρεσίας Yahoo Finance	33
3.4.2. Η βιβλιοθήκη yfinance	33
3.4.3. Δομές δεδομένων	34
3.4.4. Ενσωμάτωση στην εφαρμογή	34
3.4.5. Περιορισμοί της υπηρεσίας.....	35
4. Σχεδιασμός και υλοποίηση.....	36
4.1. Οργάνωση κώδικα.....	36
4.2. Ανάκτηση δεδομένων.....	38
4.2.1. Διαθέσιμες μετοχές	38
4.2.2. Ιστορικά δεδομένα και θεμελιώδη	39
4.3. Χρήση των DataFrames	40

4.4. Δείκτες.....	41
4.5. Στρατηγικές συναλλαγών.....	44
4.6. Προκαθορισμένες στρατηγικές	47
4.7. Διαχείριση κατάστασης.....	49
5. Αποτελέσματα και αξιολόγηση.....	51
5.1. Παραδείγματα χρήσης.....	51
5.2. Αξιολόγηση εφαρμογής και πιθανές βελτιώσεις	57
Βιβλιογραφία.....	60
Παράρτημα Α: GitHub Repository	61
Παράρτημα Β: Χρήση Τεχνητής Νοημοσύνης.....	62

Κατάλογος Εικόνων / Σχημάτων

Εικόνα 3-1: TIOBE Programming Community Index. Πηγή: www.tiobe.com	22
Εικόνα 3-2: Κατάταξη γλωσσών (δείκτης TIOBE), 06/2026. Πηγή: www.tiobe.com	22
Εικόνα 4-1: Ο κώδικας του δείκτη Moving Average.....	44
Εικόνα 4-2: Ο κώδικας της στρατηγικής Moving Average Crossover	46
Εικόνα 5-1: Stock Evaluator – Αρχική σελίδα.....	51
Εικόνα 5-2: Stock Evaluator – Επιλογή μετοχής, και διάγραμμα τιμής / όγκου	52
Εικόνα 5-3: Stock Evaluator – Προσθήκη κινητού μέσου όρου 20 ημερών	52
Εικόνα 5-4: Stock Evaluator – Προσθήκη MACD 26, 12, 9	53
Εικόνα 5-5: Stock Evaluator – Σήματα αγοράς και πώλησης.....	53
Εικόνα 5-6: Stock Evaluator – Σελίδα Screener	54
Εικόνα 5-7: Stock Evaluator – Αναζήτηση και ταξινόμηση μετοχών	55
Εικόνα 5-8: Stock Evaluator – Επιλογή μετοχών για strategy back testing	55
Εικόνα 5-9: Stock Evaluator – Σελίδα Strategies	56
Εικόνα 5-10: Stock Evaluator – Back testing στρατηγικής	57
Σχήμα 2-1: Ανοδική τάση, καθοδική τάση, και πλάγια κίνηση.....	8
Σχήμα 2-2: Παράδειγμα ανοδικής γραμμής τάσης. Πηγή: Murphy (1999, σ. 66).....	9
Σχήμα 2-3: Μετατροπή αντίστασης σε στήριξη. Πηγή: Murphy (1999, σ. 62).....	10
Σχήμα 2-4: Μετατροπή στήριξης σε αντίσταση. Πηγή: Murphy (1999, σ. 62).....	11
Σχήμα 2-5: Σχηματισμός ανερχόμενου τριγώνου. Πηγή: Murphy (1999, σ. 131)	11
Σχήμα 2-6: Τιμή κλεισίματος και κινητός μέσος 20 ημερών	13
Σχήμα 2-7: Κινητός μέσος 20 ημερών και Εκθετικός κινητός μέσος με span=20	15
Σχήμα 2-8: Momentum 14 ημερών	16
Σχήμα 2-9: RSI 14 ημερών	17
Σχήμα 2-10: Δείκτης MACD με παραμέτρους 12, 26, 9	19
Σχήμα 3-1: Λογική αρχιτεκτονική του Streamlit.....	29
Σχήμα 3-2: Κύκλος εκτέλεσης με το Streamlit	31
Σχήμα 4-1: Διάγραμμα πακέτων και εξωτερικών πηγών δεδομένων	37
Σχήμα 4-2: Διάγραμμα κλάσεων παραμετροποίησης	42
Σχήμα 4-3: Διάγραμμα κλάσεων δεικτών	43
Σχήμα 4-4: Απλοποιημένο διάγραμμα κλάσεων στρατηγικών.....	45
Σχήμα 4-5: Διάγραμμα κλάσεων στρατηγικών	48

Κατάλογος Πινάκων

Πίνακας 4-1: Δεδομένα που χρησιμοποιεί η εφαρμογή από το λεξικό Ticker.info.....	40
Πίνακας 4-1: Παράδειγμα χρήσης DataFrame στην εφαρμογή.....	41
Πίνακας 4-2: Βασικές ιδιότητες και μέθοδοι δεικτών	43
Πίνακας 4-3: Βασικές ιδιότητες και μέθοδοι στρατηγικών	46
Πίνακας 4-5: Τιμές και αντικείμενα που διατηρούνται στο session state.....	49

Συντομογραφίες & Ακρωνύμια

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ASE	Athens Stock Exchange
ASP	Active Server Pages
CSV	Comma-Separated Values
EMA	Exponential Moving Average
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JSP	Jakarta Server Pages (παλαιότερα JavaServer Pages)
LSE	London Stock Exchange
MA	Moving Average
MACD	Moving Average Convergence – Divergence
MVC	Model-View-Controller
NASDAQ	National Association of Securities Dealers Automated Quotations
NYSE	New York Stock Exchange
RSI	Relative Strength Index
SIFMA	Securities Industry and Financial Markets Association
SPA	Single-Page Application
UI	User Interface
UML	Unified Modeling Language
UX	User eXperience

1. Εισαγωγή

Οι χρηματοοικονομικές αγορές προσφέρουν ένα ευρύ φάσμα επενδυτικών προϊόντων, όπως μετοχές, ομόλογα, αμοιβαία κεφάλαια, διαπραγματεύσιμα αμοιβαία κεφάλαια, παράγωγα προϊόντα και άλλα. Μεταξύ αυτών, οι μετοχές αποτελούν ένα από τα πιο άμεσα και κατανοητά μέσα επένδυσης, καθώς με την αγορά μιας μετοχής ο επενδυτής αποκτά ένα μικρό μερίδιο ιδιοκτησίας στην εταιρεία που την έχει εκδώσει. Η απόδοση μιας μετοχής εξαρτάται από την πορεία της εταιρείας, τη διανομή μερισμάτων, τις ευρύτερες συνθήκες της αγοράς και άλλους παράγοντες, οι οποίοι μέσω του μηχανισμού της προσφοράς και της ζήτησης διαμορφώνουν καθημερινά την τιμή της.

Τα τελευταία χρόνια έχει αυξηθεί σημαντικά η συμμετοχή μικροεπενδυτών στις αγορές, εξέλιξη που συνδέεται και με την ανάπτυξη ηλεκτρονικών πλατφορμών συναλλαγών και την πρόσβαση σε αυτές μέσω Web και mobile εφαρμογών. Ενδεικτικά, το Securities Industry and Financial Markets Association (SIFMA) αναφέρει ότι το 2020 οι μικροεπενδυτές αντιστοιχούσαν περίπου στο 20%-25% του όγκου συναλλαγών στις αμερικανικές μετοχές, έναντι περίπου 10% το 2019, και αποδίδει την αύξηση αυτή σε παράγοντες όπως οι μειωμένες προμήθειες, η διευκόλυνση της πρόσβασης στις αγορές, και οι τεχνολογικές καινοτομίες στις πλατφόρμες συναλλαγών ([SIFMA, 2020](#)). Σήμερα, οι περισσότερες πλατφόρμες συναλλαγών, αλλά και αρκετά δημόσια sites και εργαλεία ανάλυσης, παρέχουν δυνατότητες που παλαιότερα ήταν διαθέσιμες μόνο σε επαγγελματίες. Οι χρήστες μπορούν να παρακολουθούν τις τιμές σε πραγματικό χρόνο ή με μικρή καθυστέρηση, να χρησιμοποιούν διαδραστικά γραφήματα, να αναλύουν μετοχές και άλλα επενδυτικά προϊόντα μέσω τεχνικών δεικτών, να συγκρίνουν προϊόντα μεταξύ τους, να ορίζουν ειδοποιήσεις και, σε ορισμένες πλατφόρμες όπως το TradingView και το MetaTrader, να δοκιμάζουν στρατηγικές συναλλαγών πάνω σε ιστορικά δεδομένα.

Παρόλο που αρκετές πλατφόρμες είναι πολύ εξελιγμένες και παρέχουν σημαντικές δυνατότητες ανάλυσης, ο χρήστης συνήθως κινείται μέσα σε προκαθορισμένα πλαίσια, δηλαδή ορίζει παραμέτρους και επιλογές για τις συγκεκριμένες λειτουργίες που είναι διαθέσιμες. Σε κάποιες περιπτώσεις, ο χρήστης μπορεί να ορίσει νέα εργαλεία, όπως τεχνικούς δείκτες, στρατηγικές συναλλαγών ή αλγόριθμους, αλλά προκειμένου να το επιτύχει χρειάζεται συνήθως να υιοθετήσει ένα ολόκληρο σύστημα κανόνων, δομών δεδομένων, ειδικής γλώσσας scripting και μοντέλου εκτέλεσης, τα οποία απαιτούν σημαντικό χρόνο εξοικείωσης και κατά κανόνα συνδρομητικά κόστη ή άνοιγμα επενδυτικού λογαριασμού.

Η παρούσα εργασία τοποθετείται ακριβώς σε αυτό το σημείο. Δεν έχει ως στόχο να ανταγωνιστεί την πληρότητα ή την ωριμότητα επαγγελματικών πλατφορμών, ούτε να χρησιμοποιηθεί για λήψη επενδυτικών αποφάσεων. Στόχος της είναι η ανάπτυξη μιας διαδικτυακής εφαρμογής στο διαδεδωμένο οικοσύστημα της Python, για πειραματισμό με αξιολόγηση μετοχών, τεχνική ανάλυση, και στρατηγικές συναλλαγών, με έμφαση όχι στα διαθέσιμα εργαλεία, αλλά στον τρόπο με τον οποίο μπορούν να υλοποιηθούν καινούργια. Για τον λόγο αυτό, η επεκτασιμότητα τέθηκε ως κεντρική, μη λειτουργική απαίτηση για την εφαρμογή, και ο σχεδιασμός της έχει γίνει με τρόπο που παρέχει έτοιμη υποδομή για την υλοποίηση νέων δεικτών και στρατηγικών συναλλαγών, με ελάχιστο και καλά οριοθετημένο κώδικα.

Στα πλαίσια της εργασίας, και ως παραδείγματα χρήσης αυτής της υποδομής, υλοποιήθηκαν βασικοί δείκτες τεχνικής ανάλυσης, όπως ο Κινητός μέσος όρος, ο Εκθετικός κινητός μέσος όρος, ο RSI, ο Momentum, και ο MACD, καθώς και διάφορες στρατηγικές συναλλαγών, τόσο απλές στρατηγικές που προκύπτουν άμεσα από τους δείκτες, όπως ανοδική ή καθοδική διάσπαση του κινητού μέσου από την τιμή, όσο και πιο σύνθετες στρατηγικές με συνδυασμούς δεικτών και πρόσθετες παραμέτρους. Η εφαρμογή παρέχει τη δυνατότητα εκτέλεσης αυτών των στρατηγικών πάνω σε ιστορικά δεδομένα (back testing) και ταξινόμησης των μετοχών με διάφορα κριτήρια απόδοσης.

Η αξία της εφαρμογής βρίσκεται στο ότι επιτρέπει συστηματικό πειραματισμό, καθαρή οργάνωση του κώδικα και δυνατότητα σταδιακής επέκτασης, στοιχεία που παρουσιάζουν ενδιαφέρον τόσο από τη σκοπιά της Τεχνολογίας Λογισμικού για εκπαιδευτικούς σκοπούς, όσο και για την βαθύτερη κατανόηση των τεχνικών δεικτών και των στρατηγικών συναλλαγών, χρησιμοποιώντας τεχνολογίες που είναι δημόσια διαθέσιμες και καλά τεκμηριωμένες.

Τα περιεχόμενα της εργασίας έχουν οργανωθεί ως εξής:

Στο Κεφάλαιο 2 (Θεωρητικό υπόβαθρο), γίνεται μια θεωρητική εισαγωγή στην ανάλυση και αξιολόγηση μετοχών και περιγράφονται οι βασικές έννοιες της θεμελιώδους και τεχνικής ανάλυσης, οι τεχνικοί δείκτες, και οι στρατηγικές συναλλαγών.

Στο Κεφάλαιο 3 (Αρχιτεκτονική και τεχνολογίες εφαρμογής) περιγράφονται οι βασικές τεχνολογικές επιλογές για την υλοποίηση της εφαρμογής, όπως η Python, το Streamlit framework, και η υπηρεσία Yahoo Finance, και τεκμηριώνεται η καταλληλότητά τους.

Το Κεφάλαιο 4 (Σχεδιασμός και υλοποίηση) περιγράφει την εφαρμογή από τη σκοπιά της Τεχνολογίας Λογισμικού, και ιδιαίτερα τις σχεδιαστικές αρχές και τα patterns που έχουν χρησιμοποιηθεί για να παραχθεί κατανοητός και εύκολα συντηρήσιμος κώδικας, και να επιτευχθεί ο βασικός στόχος της επεκτασιμότητας.

Στο Κεφάλαιο 5 (Αποτελέσματα και αξιολόγηση) γίνεται μια μικρή παρουσίαση της εφαρμογής με συγκεκριμένα παραδείγματα, αξιολογούνται τα αποτελέσματα της εργασίας, και περιγράφονται οι περιορισμοί της εφαρμογής και οι πιθανές μελλοντικές επεκτάσεις της.

2. Θεωρητικό υπόβαθρο

Ακολουθεί μια θεωρητική εισαγωγή στην ανάλυση μετοχών και ειδικότερα στην θεμελιώδη και την τεχνική ανάλυση.

Περιγράφονται συνοπτικά ορισμένα θεμελιώδη μεγέθη και δείκτες, στο βαθμό που χρησιμοποιούνται στην εφαρμογή για αναζήτηση, αξιολόγηση και ταξινόμηση μετοχών, και πιο διεξοδικά οι βασικές έννοιες και τα εργαλεία της τεχνικής ανάλυσης, όπως η τάση, τα επίπεδα στήριξης και αντίστασης και οι σχηματισμοί τιμών.

Στη συνέχεια δίνεται έμφαση στους τεχνικούς δείκτες οι οποίοι αποτελούν το κύριο αντικείμενο της εφαρμογής, καθώς προσφέρονται για αυτοματοποίηση. Οι τεχνικοί δείκτες που υλοποιούνται στην εφαρμογή παρουσιάζονται αναλυτικά, και δίνονται σχετικά παραδείγματα. Τέλος, εξηγούνται συνοπτικά τα σήματα αγοράς και πώλησης και οι στρατηγικές συναλλαγών.

2.1. Μετοχές και χρηματιστηριακά δεδομένα

Οι μετοχές αποτελούν τίτλους ιδιοκτησίας επιχειρήσεων που διατίθενται στο επενδυτικό κοινό και διαπραγματεύονται σε οργανωμένες χρηματιστηριακές αγορές, όπως το Χρηματιστήριο Αθηνών (ASE), το New York Stock Exchange (NYSE), το NASDAQ κλπ. Η τιμή μιας μετοχής μεταβάλλεται συνεχώς κατά τη διάρκεια των συνεδριάσεων, επηρεαζόμενη από οικονομικούς, επιχειρηματικούς, πολιτικούς, και ψυχολογικούς παράγοντες.

Η αξιολόγηση των μετοχών για επενδυτικούς σκοπούς πραγματοποιείται κυρίως μέσω δύο διαφορετικών προσεγγίσεων, της **Θεμελιώδους Ανάλυσης** (Fundamental Analysis) και της **Τεχνικής Ανάλυσης** (Technical Analysis).

Η θεμελιώδης ανάλυση επικεντρώνεται στην οικονομική κατάσταση και στις προοπτικές μιας εταιρείας. Στο πλαίσιο αυτό, εξετάζονται στοιχεία τα οποία αντλούνται από τις δημοσιευμένες οικονομικές καταστάσεις της εταιρείας όπως ο κύκλος εργασιών, η κερδοφορία, οι ταμειακές ροές, ο δανεισμός κλπ., καθώς και χρηματοοικονομικοί δείκτες που συνδέουν ορισμένα θεμελιώδη μεγέθη με την τρέχουσα τιμή της μετοχής, όπως τιμή προς κέρδη (Price to Earnings ή P/E)

Η θεμελιώδης ανάλυση χρησιμοποιείται συχνά για την αξιολόγηση των μακροπρόθεσμων προοπτικών μιας μετοχής, εκτίμηση της “δίκαιης αξίας” της, και λήψη επενδυτικών αποφάσεων μακροπρόθεσμου χαρακτήρα.

Αντίθετα, η τεχνική ανάλυση εστιάζει στη συμπεριφορά της αγοράς και βασίζεται κυρίως στη μελέτη του τρόπου με τον οποίο μεταβάλλονται στον χρόνο οι τιμές και οι όγκοι

συναλλαγών. Στο κλασικό έργο *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications* ([Murphy, 1999](#)), οι βασικές παραδοχές της τεχνικής ανάλυσης συνοψίζονται στο ότι: α) η αγορά προεξοφλεί τις διαθέσιμες πληροφορίες, β) οι τιμές κινούνται σε τάσεις και γ) η ιστορική συμπεριφορά τείνει να επαναλαμβάνεται. Με βάση αυτή τη λογική, ο τεχνικός αναλυτής επιχειρεί να αναγνωρίσει τάσεις και πιθανές μελλοντικές κινήσεις, χωρίς απαραίτητα να γνωρίζει τους λόγους για τους οποίους συμβαίνουν αυτές οι μεταβολές.

Οι δύο παραπάνω προσεγγίσεις χρησιμοποιούνται κατά κανόνα συμπληρωματικά. Μία συνήθης πρακτική είναι η χρήση θεμελιώδους ανάλυσης για την επιλογή των μετοχών στις οποίες θα πραγματοποιηθεί μια επένδυση και η χρήση τεχνικής ανάλυσης για τον χρονισμό (timing) αγορών και πωλήσεων.

2.2. Στοιχεία θεμελιώδους ανάλυσης

Η Θεμελιώδης Ανάλυση (Fundamental Analysis) επιχειρεί να εκτιμήσει την αξία και τις προοπτικές μιας εταιρείας με βάση τα οικονομικά της στοιχεία, τη θέση της στον κλάδο, την κερδοφορία, τη χρηματοοικονομική της διάρθρωση και τις μελλοντικές της δυνατότητες. Σε αντίθεση με την τεχνική ανάλυση που εστιάζει σε έναν τίτλο (στη μετοχή της εταιρείας), η θεμελιώδης ανάλυση αξιολογεί την ίδια την εταιρεία με βάση τα μεγέθη που μπορούν να επηρεάσουν τη μακροπρόθεσμη αξία της. Τα δεδομένα στα οποία βασίζεται η θεμελιώδης ανάλυση και ορισμένοι δείκτες που χρησιμοποιούνται κατά την αξιολόγηση μετοχών με βάση τα θεμελιώδη χαρακτηριστικά τους παρουσιάζονται συνοπτικά παρακάτω.

Η θεμελιώδης ανάλυση βασίζεται κυρίως στις οικονομικές καταστάσεις μιας εταιρείας. Από την **κατάσταση αποτελεσμάτων χρήσεως** αντλούνται στοιχεία όπως τα έσοδα, τα λειτουργικά κέρδη και τα καθαρά κέρδη. Από τον **ισολογισμό** εξετάζονται στοιχεία όπως τα περιουσιακά στοιχεία, οι υποχρεώσεις, τα ίδια κεφάλαια και ο δανεισμός. Από την **κατάσταση ταμειακών ροών** προκύπτει η ικανότητα της εταιρείας να δημιουργεί πραγματικές ταμειακές εισροές από τη λειτουργία της. Με βάση αυτά τα δεδομένα υπολογίζονται διάφοροι θεμελιώδεις δείκτες, μεταξύ των οποίων οι δείκτες αποτίμησης, που συνδέουν θεμελιώδη μεγέθη με την τρέχουσα τιμή της μετοχής. Τα στοιχεία αυτά δεν αξιολογούνται απομονωμένα, αλλά συγκριτικά με άλλες εταιρείες του ίδιου κλάδου, ή διαφορετικές χρονικές περιόδους.

Ένα βασικός δείκτης αποτίμησης είναι η **κεφαλαιοποίηση**, ή κεφαλαιοποίηση αγοράς (market capitalization). Η κεφαλαιοποίηση εκφράζει τη συνολική χρηματιστηριακή αξία μιας εταιρείας και υπολογίζεται ως το γινόμενο της τρέχουσας τιμής της μετοχής επί τον αριθμό των μετοχών που βρίσκονται σε κυκλοφορία. Παρόλο που δεν ταυτίζεται με την

“αξία” της εταιρείας, καθώς εξαρτάται από την τρέχουσα αποτίμηση της αγοράς, αποτελεί πρακτική ένδειξη του μεγέθους μιας εταιρείας. Εταιρείες μεγάλης κεφαλαιοποίησης θεωρούνται συνήθως πιο ώριμες, με μεγαλύτερη εμπορευσιμότητα και ευκολότερη πρόσβαση σε κεφάλαια, ενώ εταιρείες μικρότερης κεφαλαιοποίησης μπορεί να εμφανίζουν μεγαλύτερα περιθώρια ανάπτυξης αλλά και υψηλότερο κίνδυνο.

Άλλος σημαντικός δείκτης αποτίμησης είναι ο δείκτης **τιμής προς κέρδη**, γνωστός ως Price to Earnings ή P/E, που υπολογίζεται διαιρώντας την τιμή της μετοχής με τα κέρδη ανά μετοχή (Earnings per Share ή EPS). Με απλά λόγια, το P/E δείχνει πόσες φορές τα ετήσια κέρδη ανά μετοχή είναι διατεθειμένη να πληρώσει η αγορά για την απόκτηση της μετοχής. Ένα υψηλό P/E μπορεί να σημαίνει ότι οι επενδυτές αναμένουν υψηλότερη μελλοντική ανάπτυξη, αλλά μπορεί επίσης να αποτελεί ένδειξη ακριβής αποτίμησης. Αντίστοιχα, ένα χαμηλό P/E μπορεί να υποδηλώνει χαμηλότερη αποτίμηση, αλλά μπορεί επίσης να αντανακλά αδύναμες προοπτικές, μειωμένη εμπιστοσύνη της αγοράς ή αυξημένο επιχειρηματικό κίνδυνο. Όπως και στους περισσότερους δείκτες αποτίμησης, το τι θεωρείται υψηλό ή χαμηλό P/E εξαρτάται από το είδος της εταιρείας και τη γενικότερη κατάσταση της αγοράς, για αυτό και η αξιολόγηση του P/E γίνεται κατά κανόνα συγκριτικά με άλλες εταιρείες του ίδιου κλάδου.

Τα **κέρδη ανά μετοχή** (Earnings per Share ή EPS) δείχνουν το μέρος των καθαρών κερδών που αντιστοιχεί σε κάθε κοινή μετοχή. Αποτελούν βασικό μέγεθος κερδοφορίας και χρησιμοποιούνται σε αρκετούς δείκτες αποτίμησης, όπως το P/E. Η αύξηση των κερδών ανά μετοχή μπορεί να αποτελεί ένδειξη βελτίωσης της κερδοφορίας, αλλά η ερμηνεία του μεγέθους χρειάζεται προσοχή, επειδή μπορεί να επηρεάζεται από έκτακτα αποτελέσματα, αλλαγές στον αριθμό των μετοχών ή λογιστικούς παράγοντες.

Όταν μια εταιρεία διανέμει μερίσματα, αυτό γίνεται συνήθως σε προκαθορισμένα χρονικά διαστήματα, για παράδειγμα σε ετήσια, εξαμηνιαία ή τριμηνιαία βάση, ανάλογα με την αγορά, τη νομοθεσία και τη μερισματική πολιτική της εταιρείας. Το άθροισμα των μερισμάτων ανά μετοχή στη διάρκεια ενός έτους, εκφρασμένο ως ποσοστό επί της τιμής της μετοχής, αποτελεί τη λεγόμενη **μερισματική απόδοση** (dividend yield). Για παράδειγμα, αν μια μετοχή διαπραγματεύεται στα 10 ευρώ και το ετήσιο μέρισμα που διανέμει η εταιρεία ανά μετοχή είναι 0,40 ευρώ, η μερισματική απόδοση είναι 4%. Το στοιχείο αυτό είναι χρήσιμο για επενδυτές που ενδιαφέρονται για εισόδημα από μερίσματα και συνεκτιμάται με την προοπτική αύξησης της τιμής της μετοχής. Ωστόσο, μια υψηλή μερισματική απόδοση δεν είναι απαραίτητα θετική ένδειξη από μόνη της, επειδή μπορεί να αντανακλά αυξημένο ρίσκο ή να προκύπτει από μεγάλη πτώση της τιμής της μετοχής. Επίσης, μια εταιρεία μπορεί να μειώσει ή να διακόψει τη διανομή μερίσματος, αν

επιδεινωθούν τα οικονομικά της αποτελέσματα ή αν επιλέξει να δεσμεύσει περισσότερα κεφάλαια για επενδύσεις.

Ένας ακόμη διαδεδομένος δείκτης αποτίμησης είναι ο δείκτης **τιμής προς λογιστική αξία**, γνωστός ως Price to Book Value ή P/BV. Ο δείκτης αυτός συγκρίνει την αγοραία τιμή της μετοχής με τη λογιστική αξία ανά μετοχή, δηλαδή με την αξία των ιδίων κεφαλαίων που αντιστοιχεί σε κάθε μετοχή σύμφωνα με τον ισολογισμό. Είναι χρήσιμος σε κλάδους όπου η λογιστική αξία των περιουσιακών στοιχείων έχει μεγάλη σημασία, όπως οι τράπεζες ή οι ασφαλιστικές εταιρείες, αλλά λιγότερο αντιπροσωπευτικός για εταιρείες των οποίων η αξία βασίζεται κυρίως σε άυλα στοιχεία, όπως τεχνογνωσία, λογισμικό, εμπορικά σήματα ή δίκτυα χρηστών.

Πέρα από τους δείκτες αποτίμησης, στη θεμελιώδη ανάλυση εξετάζονται και **δείκτες κερδοφορίας, ρευστότητας και δανεισμού**, δηλαδή μεγέθη που βασίζονται αποκλειστικά στα οικονομικά στοιχεία της εταιρείας, και δεν συνδέονται άμεσα με την τιμή της μετοχής. Σχετικά παραδείγματα είναι η απόδοση ιδίων κεφαλαίων (Return on Equity ή ROE), τα περιθώρια κέρδους, ο λόγος δανεισμού προς ίδια κεφάλαια, οι δείκτες κάλυψης υποχρεώσεων, και άλλα μεγέθη που βοηθούν στην αξιολόγηση της ποιότητας και της βιωσιμότητας της κερδοφορίας, της οικονομικής ευρωστίας και του χρηματοοικονομικού κινδύνου μιας εταιρείας. Τα μεγέθη αυτά, παρότι είναι σημαντικά για τη συνολική αξιολόγηση μιας εταιρείας/μετοχής, δεν αναπτύσσονται περαιτέρω σε αυτή την εισαγωγή ούτε χρησιμοποιούνται στην εφαρμογή, καθώς ξεφεύγουν από το εύρος της εργασίας.

Όπως αναφέρθηκε ενδεικτικά για το P/E και το P/BV, η ερμηνεία των θεμελιωδών μεγεθών εξαρτάται σε μεγάλο βαθμό από τον κλάδο στον οποίο δραστηριοποιείται η εταιρεία, αλλά και τη γενικότερη κατάσταση της αγοράς. Για παράδειγμα, ένα επίπεδο δανεισμού που θεωρείται συνηθισμένο σε έναν κλάδο μπορεί να είναι υψηλό σε έναν άλλο. Αντίστοιχα, οι εταιρείες ανάπτυξης μπορεί να εμφανίζουν υψηλότερους δείκτες αποτίμησης, επειδή η αγορά προεξοφλεί μελλοντική αύξηση των κερδών, ενώ πιο ώριμες εταιρείες μπορεί να εμφανίζουν χαμηλότερη ανάπτυξη αλλά σταθερότερες ταμειακές ροές ή υψηλότερη μερισματική απόδοση. Για τον λόγο αυτό, οι θεμελιώδεις δείκτες εξετάζονται κατά κανόνα συγκριτικά, δηλαδή σε σχέση με άλλες εταιρείες του ίδιου κλάδου, ή με παλαιότερες τιμές για την ίδια εταιρεία.

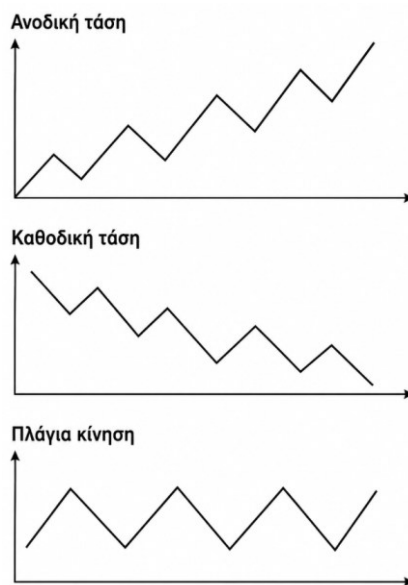
Συνοψίζοντας, η θεμελιώδης ανάλυση παρέχει ένα πλαίσιο αξιολόγησης της οικονομικής κατάστασης και της αξίας μιας εταιρείας. Στην παρούσα εργασία παρουσιάζεται μόνο σε βασικό επίπεδο, ώστε να καλυφθεί η διάσταση της αρχικής αναζήτησης και αξιολόγησης μετοχών ως επενδυτικών επιλογών. Η συνέχεια αυτού του κεφαλαίου εστιάζει στην τεχνική ανάλυση, η οποία αποτελεί τον κύριο άξονα της εργασίας.

2.3. Εισαγωγή στην τεχνική ανάλυση

Η τεχνική ανάλυση βασίζεται σε ιστορικά δεδομένα τιμών και όγκων συναλλαγών, οργανωμένα σε μορφή χρονοσειρών, είτε για μία συγκεκριμένη μετοχή, είτε για έναν χρηματιστηριακό δείκτη που αντιπροσωπεύει όλες τις μετοχές ενός κλάδου (π.χ. Τραπεζικός Δείκτης) ή μιας ολόκληρης αγοράς (π.χ. Γενικός Δείκτης Χρηματιστηρίου Αθηνών). Τα βασικά δεδομένα που χρησιμοποιούνται στην τεχνική ανάλυση είναι η τιμή ανοίγματος (Open), η μέγιστη τιμή (High), η ελάχιστη τιμή (Low), η τιμή κλεισίματος (Close), και ο όγκος συναλλαγών (Volume).

Η τεχνική ανάλυση είναι ένα αρκετά ευρύ αντικείμενο, που περιλαμβάνει την ανάγνωση διαγραμμάτων, την αναγνώριση τάσεων και επιπέδων στήριξης ή αντίστασης, τον εντοπισμό και την ερμηνεία σχηματισμών στα διαγράμματα τιμών, καθώς και την χρήση διαφόρων τεχνικών δεικτών.

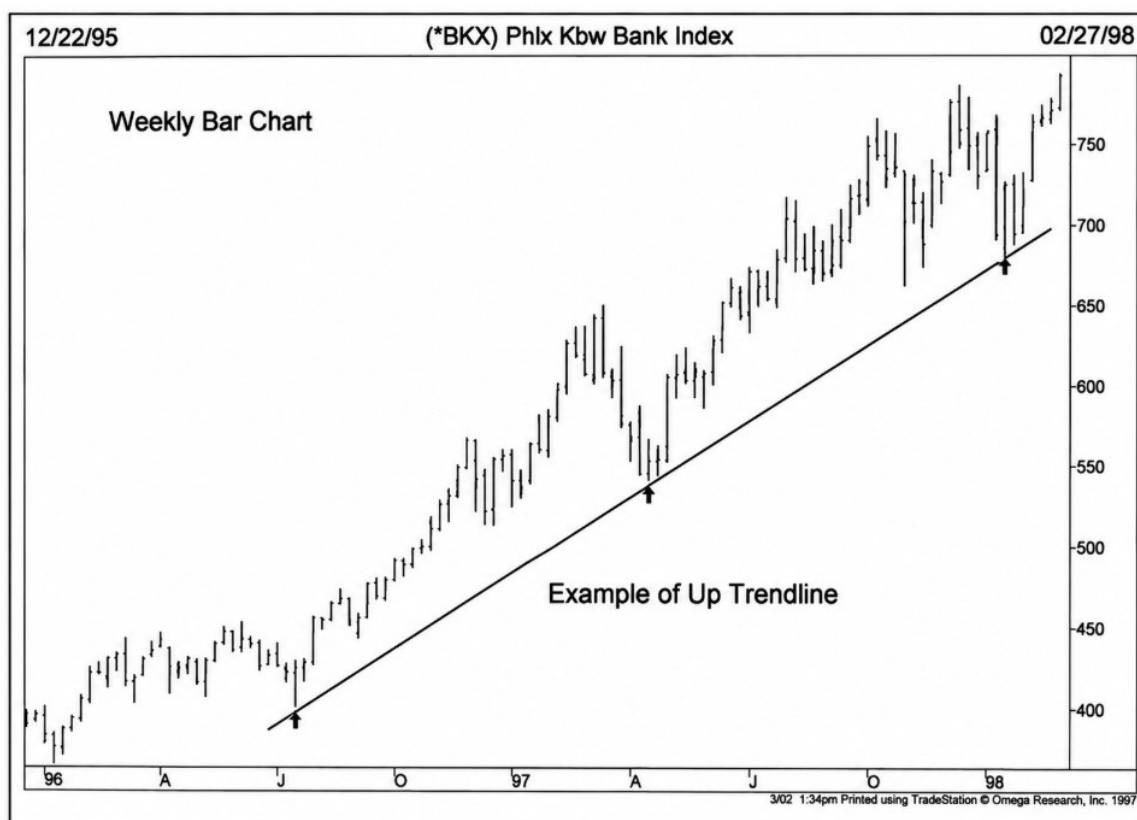
Κεντρική έννοια στην τεχνική ανάλυση είναι η **τάση**. Καθώς η τιμή μεταβάλλεται συνεχώς, σχηματίζονται τοπικά “υψηλά” και “χαμηλά”, δηλαδή σημεία στα οποία η μετοχή κινείται ανοδικά μέχρι μία μέγιστη τιμή και στη συνέχεια υποχωρεί, ή το αντίστροφο. Ως τάση νοείται η γενική κατεύθυνση προς την οποία κινείται η τιμή σε μια χρονική περίοδο, σε ευρύτερο ορίζοντα από αυτές τις τοπικές διακυμάνσεις. Η ισχυρότερη ένδειξη ανοδικής τάσης είναι ο σχηματισμός διαδοχικά υψηλότερων χαμηλών και υψηλότερων υψηλών. Αντίθετα τα διαδοχικά χαμηλότερα υψηλά και χαμηλότερα χαμηλά είναι ένδειξη καθοδικής τάσης, ενώ υπάρχουν και περιπτώσεις που η τιμή κινείται πλάγια σε σχετικά οριζόντιο εύρος, χωρίς συγκεκριμένη ανοδική ή καθοδική κατεύθυνση (Σχήμα 2-1)



Σχήμα 2-1: Ανοδική τάση, καθοδική τάση, και πλάγια κίνηση

Ακόμα και κατά τη διάρκεια ισχυρής ανόδου της τιμής συμβαίνουν “διορθώσεις”, δηλαδή προσωρινές υποχωρήσεις τις τιμές, ενώ κατά τη διάρκεια των πτώσεων μπορεί να εμφανίζονται προσωρινές ανοδικές “αντιδράσεις”. Ένας από τους λόγους που δομή των τοπικών χαμηλών και υψηλών είναι σημαντική, είναι ότι σε ανοδική τάση οι διορθώσεις τείνουν να σταματούν σε επίπεδα υψηλότερα από τα προηγούμενα χαμηλά, καθώς οι αγοραστές επανέρχονται σε υψηλότερες τιμές. Αντίστροφα, σε καθοδική τάση, οι ανοδικές αντιδράσεις τείνουν να σταματούν σε επίπεδα χαμηλότερα από τα προηγούμενα υψηλά, καθώς οι πωλητές εμφανίζονται νωρίτερα. Η αλλαγή αυτού του μοτίβου μπορεί να αποτελεί ένδειξη εξασθένησης ή πιθανής αντιστροφής της τάσης.

Στενά συνδεδεμένες με την έννοια της τάσης είναι οι **γραμμές τάσης**. Μια ανοδική γραμμή τάσης σχεδιάζεται συνήθως ενώνοντας δύο ή περισσότερα διαδοχικά χαμηλά σημεία της τιμής, ενώ μια καθοδική γραμμή τάσης σχεδιάζεται ενώνοντας δύο ή περισσότερα διαδοχικά υψηλά σημεία. Η γραμμή αυτή λειτουργεί ως οπτικό εργαλείο που βοηθά τον αναλυτή να εκτιμήσει τη γενική κατεύθυνση της αγοράς και τα πιθανά σημεία όπου η τιμή μπορεί να αντιδράσει. Όσο περισσότερες φορές η τιμή προσεγγίζει μια γραμμή τάσης χωρίς να τη διασπά, τόσο μεγαλύτερη σημασία αποκτά η γραμμή αυτή για την τεχνική ανάλυση. Ένα χαρακτηριστικό παράδειγμα ανοδικής γραμμής τάσης δίνεται στο Σχήμα 2-2.

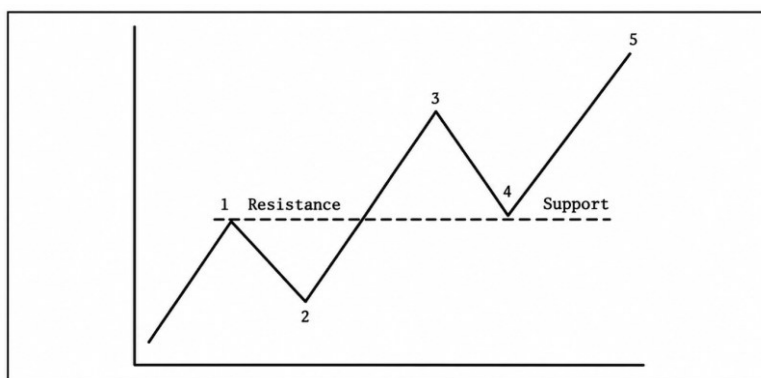


Σχήμα 2-2: Παράδειγμα ανοδικής γραμμής τάσης. Πηγή: Murphy (1999, σ. 66)

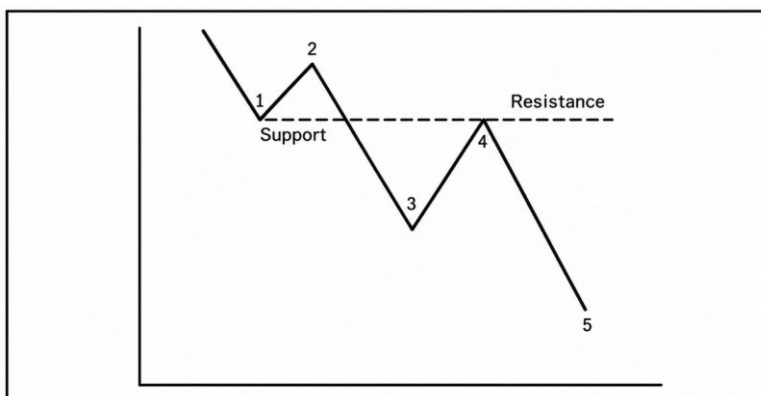
Η διάσπαση μιας γραμμής τάσης θεωρείται σημαντικό γεγονός, επειδή μπορεί να δείχνει αλλαγή στη συμπεριφορά της αγοράς. Η διάσπαση, ωστόσο, δεν θεωρείται πάντοτε επαρκές στοιχείο από μόνη της. Συνήθως εξετάζεται σε συνδυασμό με τον όγκο συναλλαγών, τη συμπεριφορά της τιμής στις επόμενες συνεδριάσεις, και άλλους τεχνικούς δείκτες.

Μία ακόμη βασική έννοια στην τεχνική ανάλυση είναι τα επίπεδα **στήριξης (support)** και **αντίστασης (resistance)**. Στήριξη θεωρείται ένα επίπεδο τιμών στο οποίο η ζήτηση εμφανίζεται αρκετά ισχυρή ώστε να ανακόπτει ή να επιβραδύνει την πτώση της τιμής, ενώ αντίσταση θεωρείται ένα επίπεδο στο οποίο η προσφορά εμφανίζεται αρκετά ισχυρή ώστε να ανακόπτει ή να επιβραδύνει την άνοδο. Τα επίπεδα αυτά προκύπτουν συνήθως από προηγούμενα υψηλά ή χαμηλά της τιμής, δηλαδή από περιοχές όπου στο παρελθόν υπήρξε σημαντική αντίδραση της αγοράς, για λόγους που σχετίζονται με τη μνήμη της αγοράς, την ψυχολογία και τη συμπεριφορά των επενδυτών. Ένα επίπεδο όπου στο παρελθόν εμφανίστηκαν αγοραστές είναι πιθανό σημείο νέας αγοραστικής αντίδρασης. Αντίστοιχα, ένα επίπεδο όπου στο παρελθόν εμφανίστηκαν πωλητές μπορεί να λειτουργήσει ξανά ως εμπόδιο στην άνοδο.

Πολύ συχνά, όταν ένα σημαντικό επίπεδο διασπαστεί ο ρόλος του αντιστρέφεται. Δηλαδή, μια προηγούμενη αντίσταση που διασπάται ανοδικά αρχίζει να λειτουργεί ως στήριξη (Σχήμα 2-3), ενώ μια προηγούμενη στήριξη που διασπάται καθοδικά αρχίζει να λειτουργεί ως αντίσταση (Σχήμα 2-4).



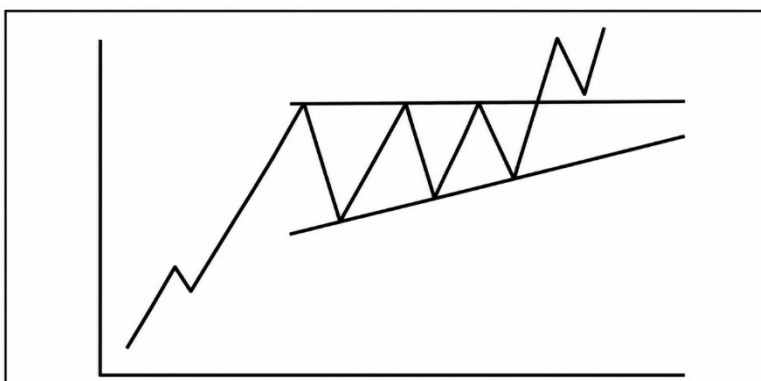
Σχήμα 2-3: Μετατροπή αντίστασης σε στήριξη. Πηγή: Murphy (1999, σ. 62)



Σχήμα 2-4: Μετατροπή στήριξης σε αντίσταση. Πηγή: Murphy (1999, σ. 62)

Αυτή η εναλλαγή ρόλων εμφανίζεται συχνά στην τεχνική ανάλυση και συνδέεται με την αλλαγή προσδοκιών στην αγορά.

Πέρα από τις γραμμές τάσης και τα επίπεδα στήριξης και αντίστασης, πρέπει να αναφερθεί επιγραμματικά ότι η τεχνική ανάλυση εξετάζει και **σχηματισμούς τιμών**, δηλαδή επαναλαμβανόμενα μοτίβα που εμφανίζονται στα διαγράμματα και θεωρείται ότι εκφράζουν συγκεκριμένη δυναμική μεταξύ αγοραστών και πωλητών ή παρέχουν ενδείξεις **συνέχειας** ή **αντιστροφής** της τάσης. Ένα απλό παράδειγμα σχηματισμού συνέχειας είναι το ascending triangle (ανοδικό τρίγωνο), στο οποίο μια γενικά ανοδική τάση συναντά ένα επίπεδο αντίστασης και εμφανίζει μια σειρά ταλαντώσεων με προοδευτικά μειούμενο εύρος. Η δημιουργία ανοδικού τριγώνου είναι μια ένδειξη ότι οι αγοραστές είναι πιο “επιθετικοί” από τους πωλητές, για αυτό και κατά κανόνα το επίπεδο αντίστασης τελικά διασπάται πριν την εξάντληση του τριγώνου που σχηματίζεται. Μετά τη διάσπαση, τυχόν διόρθωση που αποτυγχάνει να επαναφέρει τη τιμή κάτω από αυτό το επίπεδο αποτελεί ακόμα πιο ισχυρή ένδειξη για συνέχεια της ανόδου (Σχήμα 2-5).



Σχήμα 2-5: Σχηματισμός ανερχόμενου τριγώνου. Πηγή: Murphy (1999, σ. 131)

Σημειώνεται ότι ο **όγκος συναλλαγών** έχει υποστηρικτικό αλλά σημαντικό ρόλο στην τεχνική ανάλυση. Η τιμή δείχνει την κατεύθυνση της κίνησης, ενώ ο όγκος δείχνει την

ένταση της κίνησης, ή το βαθμό συμμετοχής αγοραστών και πωλητών σε αυτήν την κίνηση. Για παράδειγμα, μια διάσπαση ενός επιπέδου αντίστασης με μικρό όγκο συναλλαγών δεν αξιολογείται ως ισχυρή ένδειξη ανόδου και συχνά ακολουθείται από νέα πτώση.

Τα εργαλεία και οι τεχνικές που περιγράφηκαν μέχρι τώρα είναι χρήσιμα για την κατανόηση των βασικών εννοιών της τεχνικής ανάλυσης, ωστόσο έχουν ένα σημαντικό μειονέκτημα που επισημαίνει και ο ίδιος ο Murphy: η ανάλυση των διαγραμμάτων με αυτές τις μεθόδους είναι αρκετά υποκειμενική και είναι δύσκολο να αυτοματοποιηθεί και να ελεγχθεί πειραματικά. Η δυσκολία αυτή αίρεται με τη χρήση των **τεχνικών δεικτών**, οι οποίοι ορίζονται μέσω συγκεκριμένων μαθηματικών τύπων, και επομένως προσφέρονται περισσότερο για προγραμματιστική υλοποίηση και αυτοματοποιημένο έλεγχο πάνω σε ιστορικά δεδομένα.

Οι τεχνικοί δείκτες (που στο εξής αναφέρονται και ως δείκτες τεχνικής ανάλυσης, ή απλώς δείκτες), καθώς και η υλοποίηση επενδυτικών στρατηγικών με βάση αυτούς τους δείκτες, είναι από τα σημαντικότερα κεφάλαια της τεχνικής ανάλυσης, και οι τομείς στους οποίους επικεντρώνεται η παρούσα εργασία. Τα θέματα αυτά περιγράφονται αναλυτικά στις επόμενες παραγράφους.

2.4. Δείκτες τεχνικής ανάλυσης

Οι δείκτες είναι διάφοροι μαθηματικοί ή στατιστικοί υπολογισμοί που εκτελούνται πάνω στα ιστορικά δεδομένα των τιμών, με σκοπό να αναλυθεί η συμπεριφορά της αγοράς και να αναγνωριστούν χαρακτηριστικά όπως η γενική τάση της αγοράς, η ορμή των μεταβολών (momentum), η μεταβλητότητα, καθώς και πιθανές συνθήκες υπερβολής (π.χ. «υπεραγορασμένη» ή «υπερπωλημένη» μετοχή).

Στους περισσότερους δείκτες, όπως και στα υπόλοιπα εργαλεία της τεχνικής ανάλυσης που περιγράφηκαν στην προηγούμενη παράγραφο, χρησιμοποιείται συνήθως η τιμή κλεισίματος (close) η οποία θεωρείται ιδιαίτερα σημαντική και αντιμετωπίζεται ως η αντιπροσωπευτικότερη τιμή της συνεδρίασης.

Στο έργο του Murphy ([1999](#)) παρουσιάζεται μεγάλος αριθμός δεικτών τεχνικής ανάλυσης, καθώς και οι βασικές αρχές χρήσης και ερμηνείας τους. Οι δείκτες αυτοί μπορούν να διαχωριστούν σε διάφορες κατηγορίες, όπως δείκτες τάσης (trend indicators), δείκτες ορμής (momentum indicators) και ταλαντωτές (oscillators).

Οι βασικότεροι τεχνικοί δείκτες εξηγούνται συνοπτικά στη συνέχεια. Όλα τα διαγράμματα που παρουσιάζονται έχουν δημιουργηθεί με screenshots από το **Stock Evaluator**, την εφαρμογή που αναπτύχθηκε στα πλαίσια της εργασίας.

2.4.1. Κινητός μέσος όρος

Ο απλούστερος και πιο διαδεδομένος δείκτης τεχνικής ανάλυσης είναι ο Κινητός Μέσος Όρος ή Moving Average (MA). Ο δείκτης αυτός υπολογίζει τη μέση τιμή μιας μετοχής για έναν συγκεκριμένο αριθμό προηγούμενων συνεδριάσεων, συνήθως με βάση τις τιμές κλεισίματος. Για παράδειγμα, ένας κινητός μέσος όρος 20 ημερών, που αναφέρεται και ως MA20, δείχνει τη μέση τιμή κλεισίματος των τελευταίων 20 συνεδριάσεων.

Ενδεικτικά, στο Σχήμα 2-6 εμφανίζεται το διάγραμμα της τιμής και του κινητού μέσου όρου 20 ημερών της μετοχής της Aegean Airlines για το τελευταίο έτος. Όπως φαίνεται καθαρά στο διάγραμμα, ο κινητός μέσος όρος απεικονίζει μια εξομαλυμένη εικόνα για την πορεία της τιμής στον χρόνο. Με αυτόν τον τρόπο μειώνει την επίδραση των καθημερινών διακυμάνσεων και κάνει ευκολότερη την αναγνώριση της γενικής κατεύθυνσης της αγοράς.



Σχήμα 2-6: Τιμή κλεισίματος και κινητός μέσος 20 ημερών

Η περίοδος των 20 ημερών θεωρείται ενδεικτική της τρέχουσας κατάστασης, ενώ μια περίοδος π.χ. 200 ημερών θεωρείται ενδεικτική της πιο μακροπρόθεσμης τάσης. Οι τιμές 20 και 200 είναι ιδιαίτερα διαδεδομένες, αλλά η επιλογή της κατάλληλης περιόδου εξαρτάται από πολλούς παράγοντες, όπως ο χρονικός ορίζοντας της ανάλυσης, η συχνότητα συναλλαγών, και η μεταβλητότητα της μετοχής.

Όταν η τιμή κινείται πάνω από τον κινητό μέσο, ενδεικτικά των 20 ημερών, δηλαδή όταν η τρέχουσα αποτίμηση της μετοχής είναι υψηλότερη από τον πρόσφατο μέσο όρο της, αυτό υποδεικνύει ότι υπάρχει ανοδική δυναμική. Αντίθετα, όταν η τιμή κινείται κάτω από τον κινητό μέσο, αυτό μπορεί να θεωρηθεί ένδειξη καθοδικής πίεσης. Για το λόγο αυτό, στα διαγράμματα τιμής – κινητού μέσου όρου, έχει ιδιαίτερη σημασία η ανοδική ή καθοδική “διάσπαση” της γραμμής του δείκτη από τη γραμμή της τιμής, δηλαδή η μετάβαση της τιμής σε επίπεδο υψηλότερο ή χαμηλότερο από τον πρόσφατο μέσο όρο, καθώς αυτό υποδεικνύει πιθανή αλλαγή στη συμπεριφορά της μετοχής.

2.4.2. Εκθετικός κινητός μέσος όρος

Εναλλακτικά, αντί του απλού κινητού μέσου όρου, χρησιμοποιείται ο Εκθετικός Κινητός Μέσος (Exponential Moving Average ή EMA). Σε αυτόν το δείκτη, λαμβάνονται υπόψη όλες οι ιστορικές τιμές της μετοχής και όχι μόνο οι τελευταίες n ημέρες, αλλά οι πιο πρόσφατες τιμές έχουν μεγαλύτερη βαρύτητα από τις παλαιότερες (η επίδραση των παλαιότερων τιμών τείνει ασυμπτωτικά στο μηδέν). Με αυτόν τον τρόπο ο δείκτης παρουσιάζει μικρότερη υστέρηση από τον απλό κινητό μέσο, δηλαδή αντιδρά ταχύτερα στις μεταβολές της τιμής, ενώ εξακολουθεί να εξομαλύνει τις βραχυπρόθεσμες διακυμάνσεις. Ο αναδρομικός τύπος του εκθετικού κινητού μέσου είναι:

$$EMA_t = \alpha \cdot P_t + (1 - \alpha) \cdot EMA_{t-1}$$

όπου P_t είναι η τιμή της τρέχουσας ημέρας, EMA_{t-1} είναι η τιμή του δείκτη την προηγούμενη ημέρα και α είναι ένας συντελεστής εξομάλυνσης, ο οποίος καθορίζει πόσο βάρος δίνεται στην πιο πρόσφατη τιμή. Στην πράξη, ο εκθετικός κινητός μέσος παραμετροποιείται συνήθως μέσω της παραμέτρου “span” με την οποία υπολογίζεται ο συντελεστής εξομάλυνσης ως εξής:

$$\alpha = \frac{2}{\text{span} + 1}$$

Για να γίνει κατανοητή η φυσική σημασία της παραμέτρου span, σημειώνεται ότι ένας απλός κινητός μέσος n ημερών παρουσιάζει κατά προσέγγιση την ίδια υστέρηση στις μεταβολές της τιμής με έναν εκθετικό κινητό μέσο με $\text{span}=n$, δηλαδή επηρεάζεται σε παρόμοιο βαθμό από τις παλαιότερες τιμές. Για παράδειγμα, η σταθμισμένη “μέση ηλικία” των τιμών που επηρεάζουν έναν EMA με $\text{span}=20$ είναι περίπου 10 ημέρες, ίδια με τη μέση ηλικία των τιμών στον απλό MA 20 ημερών. Η σχέση μεταξύ των δύο δεικτών φαίνεται και στο παράδειγμα του επόμενου διαγράμματος, που δείχνει την τιμή της μετοχής της Εθνικής Τράπεζας (ETE) και τους δείκτες MA20 και EMA20 για το τελευταίο εξάμηνο (Σχήμα 2-7).



Σχήμα 2-7: Κινητός μέσος 20 ημερών και Εκθετικός κινητός μέσος με span=20

2.4.3. Δείκτης ορμής (Momentum)

Ο Δείκτης ορμής, ή Momentum, είναι ένας απλός ταλαντωτής (oscillator). Σε αντίθεση με τους κινητούς μέσους, οι ταλαντωτές μεταβάλλονται είτε μέσα σε προκαθορισμένο εύρος τιμών είτε γύρω από μια κεντρική γραμμή αναφοράς, συνήθως το μηδέν. Επειδή η κλίμακά τους είναι διαφορετική από την κλίμακα της τιμής της μετοχής, προβάλλονται κατά κανόνα σε ξεχωριστό διάγραμμα, κάτω από το βασικό διάγραμμα της τιμής.

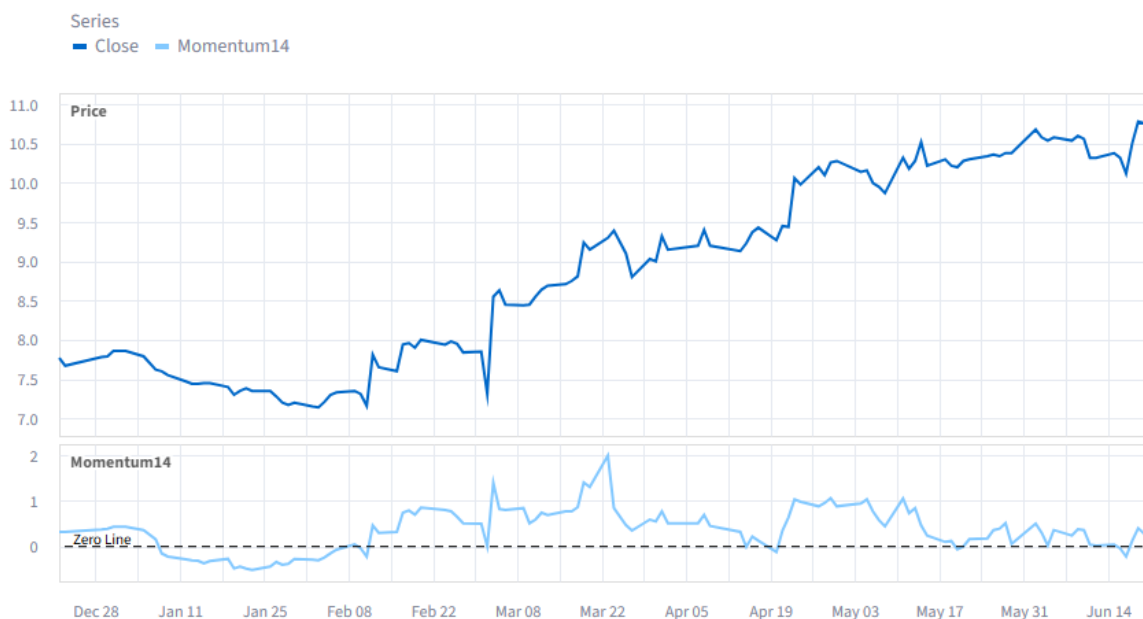
Ο Momentum συγκρίνει την τρέχουσα τιμή με την τιμή που είχε η μετοχή πριν από έναν συγκεκριμένο αριθμό ημερών, και δίνει ένα μέτρο της ταχύτητας και της κατεύθυνσης με την οποία μεταβάλλεται η τιμή. Παραμετροποιείται με τον αριθμό των ημερών (n) και υπολογίζεται με βάση τη σχέση:

$$\text{Momentum}_t = P_t - P_{t-n}$$

όπου P_t είναι η τιμή της τρέχουσας ημέρας και P_{t-n} είναι η τιμή πριν από n ημέρες. Όταν η τιμή του δείκτη είναι θετική (αρνητική), η τρέχουσα τιμή είναι υψηλότερη (χαμηλότερη) από πριν, γεγονός που δείχνει ανοδική (καθοδική) δυναμική. Τιμή κοντά στο μηδέν σημαίνει ότι η τιμή της μετοχής δεν έχει μεταβληθεί σημαντικά κατά το επιλεγμένο χρονικό διάστημα.

Στην πράξη, ο Momentum χρησιμοποιείται κυρίως για να εκτιμηθεί αν η πρόσφατη κίνηση της μετοχής ενισχύεται ή εξασθενεί. Για παράδειγμα, αύξηση του Momentum σε θετικές τιμές δείχνει ενίσχυση της ανοδικής κίνησης, ενώ μείωση του Momentum ή μετάβασή του σε αρνητικές τιμές μπορεί να αποτελεί ένδειξη εξασθένησης της τάσης ή πιθανής αντιστροφής. Όπως και οι υπόλοιποι ταλαντωτές, ο Momentum δεν χρησιμοποιείται συνήθως ως αυτόνομο σήμα αγοράς ή πώλησης, αλλά σε συνδυασμό με άλλους δείκτες ή κανόνες στρατηγικής.

Στο παρακάτω διάγραμμα (Σχήμα 2-8), παρουσιάζεται ενδεικτικά η τιμή της μετοχής της ΕΥΔΑΠ για το τελευταίο εξάμηνο, μαζί με τον δείκτη Momentum 14 ημερών.



Σχήμα 2-8: Momentum 14 ημερών

2.4.4. Δείκτης σχετικής ισχύος (RSI)

Ένας ακόμη βασικός δείκτης τεχνικής ανάλυσης είναι ο Δείκτης Σχετικής Ισχύος (Relative Strength Index ή RSI), ο οποίος ανήκει και αυτός στην κατηγορία των ταλαντωτών (oscillators). Ο RSI κινείται μέσα σε προκαθορισμένο εύρος τιμών, από 0 έως 100, και χρησιμοποιείται για να εκτιμηθεί η ένταση των πρόσφατων ανοδικών και καθοδικών κινήσεων της μετοχής. Ο δείκτης παραμετροποιείται με έναν αριθμό ημερών, συνήθως 14, και υπολογίζεται με βάση τη σχέση:

$$RSI = 100 - \frac{100}{1 + RS}$$

όπου RS είναι ο λόγος της μέσης ανοδικής μεταβολής προς τη μέση καθοδική μεταβολή της τιμής για τις τελευταίες n ημέρες, εκφρασμένες ως ποσοστά:

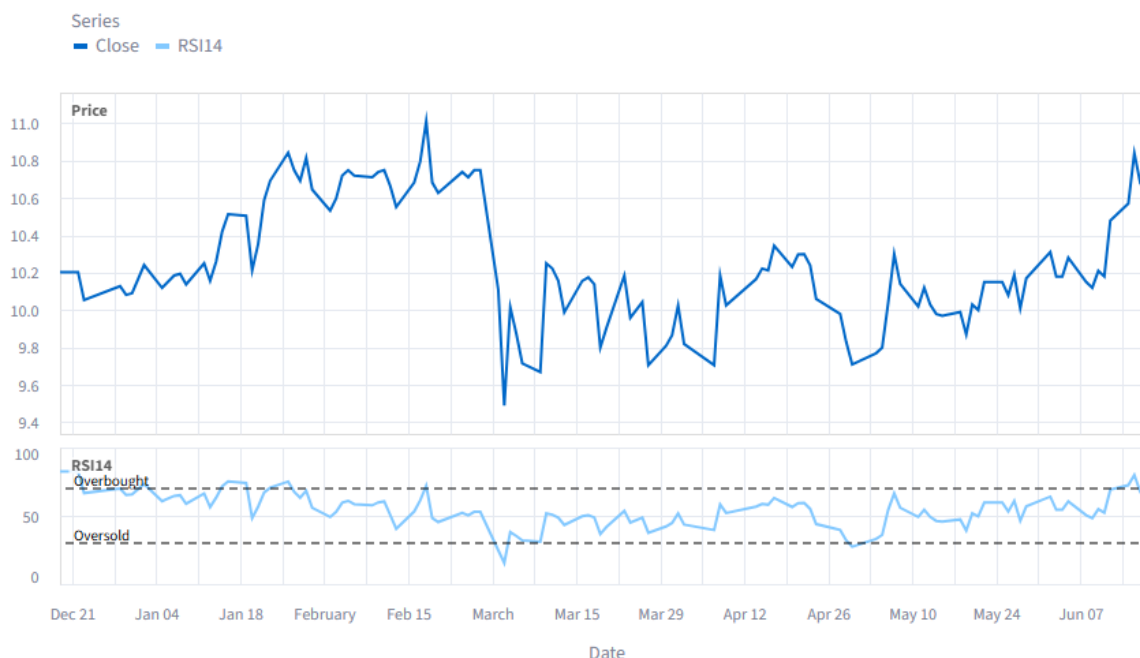
$$RS = \frac{\text{μέση ανοδική μεταβολή } n \text{ ημερών (\%)}}{\text{μέση καθοδική μεταβολή } n \text{ ημερών (\%)}}$$

Από τους παραπάνω τύπους είναι προφανές ότι σε εξαιρετικά ισχυρή και απότομη άνοδο ή πτώση, ο δείκτης κινείται προς το 100 ή το 0, αντίστοιχα. Στην πράξη, όταν ο RSI 14 ημερών ξεπεράσει ένα επίπεδο (συνήθως το 70) θεωρείται ότι η απότομη άνοδος της τιμής αγγίζει τα όρια της υπερβολής και η μετοχή θεωρείται “overbought” (υπεραγορασμένη).

Αντίστροφα, όταν ο RSI 14 ημερών πέσει κάτω από ένα επίπεδο (συνήθως 30) η μετοχή θεωρείται “oversold” (υπερπωλημένη).

Τα όρια αυτά δεν αποτελούν από μόνα τους σήματα αγοράς ή πώλησης, αλλά ενδείξεις πιθανής εξασθένησης ή αντιστροφής της πρόσφατης κίνησης. Πολύ συχνά, ο RSI χρησιμοποιείται συνδυαστικά, για παράδειγμα ως πρόσθετο κριτήριο επιβεβαίωσης ενός άλλου ανοδικού ή καθοδικού σήματος, ή για τον ακριβή χρονισμό μιας συναλλαγής που έχει αποφασιστεί με άλλα κριτήρια.

Στο παρακάτω διάγραμμα (Σχήμα 2-9), παρουσιάζεται ενδεικτικά η τιμή της μετοχής του Διεθνούς Αερολιμένα Αθηνών (ΑΙΑ) για το τελευταίο εξάμηνο, μαζί με τον δείκτη RSI 14 ημερών. Επισημαίνεται η εκτεταμένη είσοδος του RSI στην υπερπωλημένη περιοχή κατά την απότομη πτώση της τιμής της μετοχής στα τέλη Φεβρουαρίου και τις αρχές Μαρτίου.



Σχήμα 2-9: RSI 14 ημερών

2.4.5. Δείκτης σύγκλισης – απόκλισης κινητών μέσων (MACD)

Ο δείκτης Σύγκλισης – Απόκλισης Κινητών Μέσων (Moving Average Convergence – Divergence ή MACD) είναι ένας δείκτης που συνδυάζει στοιχεία δείκτη τάσης και ταλαντωτή. Βασίζεται στη διαφορά δύο εκθετικών κινητών μέσων διαφορετικού μήκους, συνήθως ενός “γρήγορου” εκθετικού μέσου 12 ημερών και ενός “αργού” εκθετικού μέσου 26 ημερών. Ο γρήγορος κινητός μέσος αντιδρά ταχύτερα στις μεταβολές της τιμής, ενώ ο αργός κινητός μέσος ακολουθεί πιο ομαλά τη γενικότερη πορεία της μετοχής.

Ο βασικός τύπος του δείκτη είναι:

$$\text{MACD} = \text{EMA}_{\text{short}} - \text{EMA}_{\text{long}}$$

όπου $\text{EMA}_{\text{short}}$ είναι ο εκθετικός κινητός μέσος μικρότερου μήκους και EMA_{long} είναι ο εκθετικός κινητός μέσος μεγαλύτερου μήκους. Όταν η τιμή του MACD είναι θετική, ο γρήγορος κινητός μέσος βρίσκεται πάνω από τον αργό, γεγονός που δείχνει ανοδική δυναμική, και το αντίθετο όταν η τιμή του MACD είναι αρνητική.

Μαζί με τον MACD υπολογίζεται και η λεγόμενη **γραμμή σήματος** (signal line), η οποία είναι ο εκθετικός κινητός μέσος του ίδιου του MACD:

$$\text{Signal} = \text{EMA}_{\text{trigger}}(\text{MACD})$$

όπου trigger είναι η περίοδος αυτού του εκθετικού μέσου, συνήθως 9 ημέρες. Η γραμμή σήματος εξομαλύνει τον MACD και χρησιμοποιείται ως γραμμή αναφοράς για την ερμηνεία των μεταβολών του δείκτη, παρόμοια με τη χρήση του κινητού μέσου σε σχέση με την τιμή. Όταν ο MACD διασπά ανοδικά τη γραμμή σήματος, αυτό θεωρείται ένδειξη ενίσχυσης της ανοδικής δυναμικής. Αντίθετα, όταν ο MACD διασπά καθοδικά τη γραμμή σήματος, αυτό θεωρείται ένδειξη ενίσχυσης της καθοδικής δυναμικής.

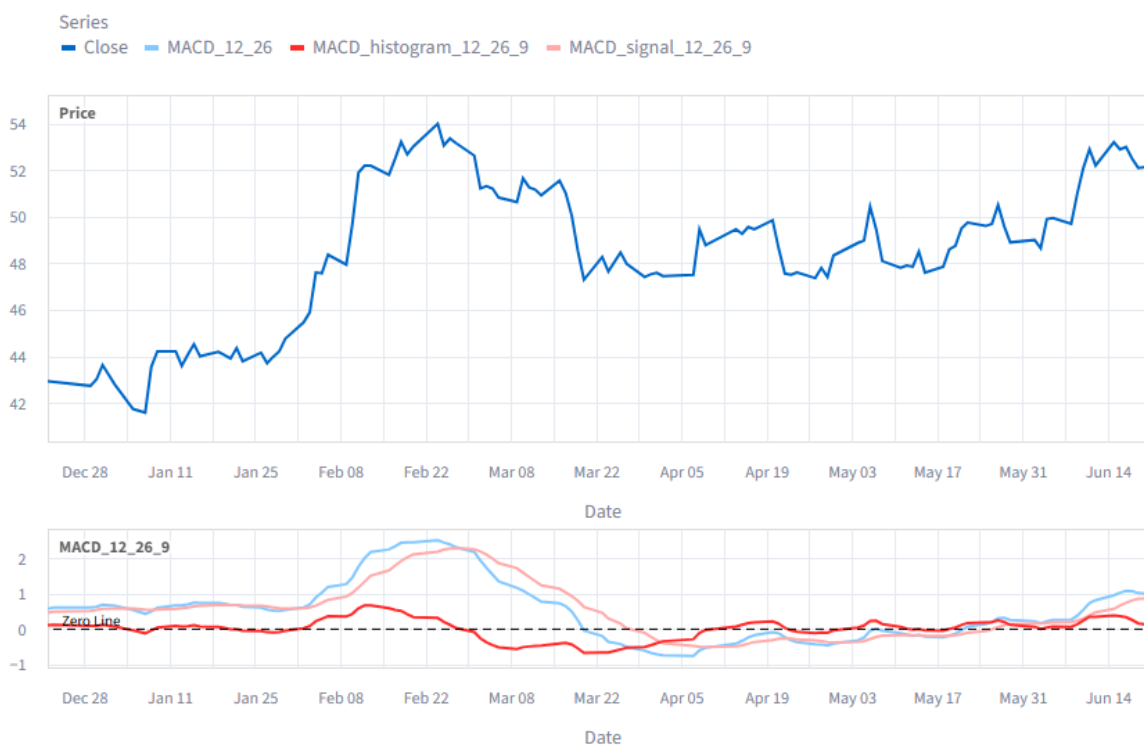
Επιπλέον, υπολογίζεται και το λεγόμενο **ιστόγραμμα** του MACD, το οποίο εκφράζει τη διαφορά μεταξύ του MACD και της γραμμής σήματος:

$$\text{Histogram} = \text{MACD} - \text{Signal}$$

Το ιστόγραμμα μεταβάλλεται γύρω από το μηδέν, και δείχνει πόσο απομακρύνεται ο MACD από τη γραμμή σήματος. Όταν το ιστόγραμμα είναι θετικό, ο MACD βρίσκεται πάνω από τη γραμμή σήματος, ενώ όταν είναι αρνητικό βρίσκεται κάτω από αυτήν. Η αύξηση του ιστογράμματος δείχνει ότι η απόσταση μεταξύ των δύο γραμμών μεγαλώνει, άρα η αντίστοιχη δυναμική ενισχύεται, ενώ η μείωσή του δείχνει ότι η δυναμική εξασθενεί.

Ο MACD παραμετροποιείται με τις τρεις περιόδους που απαιτούνται για τον υπολογισμό του (γρήγορος μέσος, αργός μέσος, και περίοδος σήματος) και στη διαγραμματική του απεικόνιση εμφανίζει τις τρεις γραμμές που περιγράφηκαν (MACD, Signal, και Histogram). Όπως και οι υπόλοιποι ταλαντωτές, ο MACD προβάλλεται σε ξεχωριστό διάγραμμα κάτω από το διάγραμμα της τιμής, και για αναγνωσιμότητα, αλλά και επειδή η κλίμακά του είναι διαφορετική από την κλίμακα της τιμής της μετοχής.

Ενδεικτικά, το Σχήμα 2-10 δείχνει την τιμή και τον MACD 26, 12, 9 της μετοχής Coca Cola HBC AG (EEE) για το τελευταίο εξάμηνο.



Σχήμα 2-10: Δείκτης MACD με παραμέτρους 12, 26, 9

2.5. Σήματα και στρατηγικές συναλλαγών

Με βάση όσα αναφέρθηκαν στις προηγούμενες παραγράφους, και χρησιμοποιώντας ως παράδειγμα τον κινητό μέσο όρο 20 ημερών, η ανοδική διάσπαση του δείκτη από την τιμή μπορεί να θεωρηθεί **σήμα αγοράς** (buy signal), δηλαδή ένδειξη ότι η μετοχή πρόκειται να κινηθεί ψηλότερα. Αντίστροφα, η καθοδική διάσπαση του κινητού μέσου όρου μπορεί να θεωρηθεί ένα **σήμα πώλησης** (sell signal).

Στην πράξη, τα σήματα αγοράς και πώλησης ή **σήματα συναλλαγών** (trade signals) σπάνια χρησιμοποιούνται χωρίς πρόσθετους κανόνες, αφενός επειδή σε περιόδους πλάγιας κίνησης ή υψηλής μεταβλητότητας μπορεί να δημιουργούνται πολλά εσφαλμένα σήματα, και αφετέρου επειδή η στιγμιαία επαλήθευση ενός και μόνο κριτηρίου δεν θεωρείται πάντα ικανή συνθήκη για την αναγνώριση ενός έγκυρου σήματος, οπότε χρειάζεται επαλήθευση ή διασταύρωση με πρόσθετα κριτήρια. Για παράδειγμα, ο τεχνικός αναλυτής μπορεί να θεωρήσει ότι μια ανοδική διάσπαση του κινητού μέσου όρου 20 ημερών είναι έγκυρη μόνο αν η τιμή κλεισίματος παραμείνει πάνω από τον κινητό μέσο όρο και στην επόμενη συνεδρίαση. Αυτό είναι ένα απλό **φίλτρο** επιβεβαίωσης δύο ημερών το οποίο μειώνει την πιθανότητα αποδοχής βραχυπρόθεσμων ή τυχαίων διασπάσεων. Άλλα φίλτρα που χρησιμοποιούνται συχνά είναι η εφαρμογή ενός ελάχιστου αριθμού ημερών μετά από ένα σήμα πριν αναγνωριστούν νέα σήματα, καθώς και η χρήση ενός μακροπρόθεσμου κινητού

μέσου όρου ως φίλτρου τάσης, για παράδειγμα επικύρωση σημάτων αγοράς μόνο αν η τιμή βρίσκεται πάνω από τον κινητό μέσο όρο 200 ημερών.

Το σύνολο των κανόνων που καθορίζουν πότε δημιουργούνται σήματα αγοράς και πώλησης και υπό ποιες προϋποθέσεις αυτά γίνονται αποδεκτά ονομάζεται **στρατηγική συναλλαγών**. Το παράδειγμα της διάσπασης ενός κινητού μέσου όρου είναι σχετικά απλό και βασίζεται σε έναν μόνο δείκτη. Πιο σύνθετες στρατηγικές μπορεί να βασίζονται σε περισσότερους δείκτες ή σε συνδυασμούς δεικτών. Για παράδειγμα, στην στρατηγική διασταύρωσης κινητών μέσων όρων (Moving Average Crossover), δεν συγκρίνεται η τιμή με έναν κινητό μέσο όρο, αλλά δύο κινητοί μέσοι όροι διαφορετικής διάρκειας μεταξύ τους. Σε μια τέτοια προσέγγιση, η ανοδική διάσπαση ενός μακροπρόθεσμου κινητού μέσου όρου (π.χ. 200 ημερών) από έναν βραχυπρόθεσμο (π.χ. 50 ημερών) μπορεί να ερμηνευθεί ως ένδειξη ενίσχυσης της ανοδικής τάσης, ενώ η αντίστροφη διάσπαση μπορεί να ερμηνευθεί ως ένδειξη εξασθένησης ή αντιστροφής της τάσης.

2.6. Επισημάνσεις

Η αποτελεσματικότητα της τεχνικής ανάλυσης έχει αποτελέσει αντικείμενο αρκετών εμπειρικών μελετών, χωρίς να υπάρχει απόλυτη συμφωνία στη βιβλιογραφία. Ορισμένες κλασικές μελέτες, όπως των Brock et al. ([1992](#)), έχουν δείξει ότι απλοί κανόνες τεχνικής ανάλυσης, όπως οι κινητοί μέσοι όροι, παράγουν στατιστικά σημαντικά αποτελέσματα σε ιστορικά δεδομένα. Από την άλλη πλευρά, επισκοπήσεις της βιβλιογραφίας, όπως αυτή των Park και Irwin ([2007](#)), δείχνουν ότι τα αποτελέσματα διαφέρουν ανάλογα με την αγορά, την περίοδο, τη μεθοδολογία ελέγχου και το είδος των τεχνικών κανόνων που εξετάζονται.

Συνεπώς, η τεχνική ανάλυση δεν μπορεί να θεωρηθεί γενικά αποδεδειγμένη ή πάντοτε κερδοφόρα μέθοδος πρόβλεψης των τιμών. Παρόλα αυτά, αρκετές μελέτες δείχνουν ότι συγκεκριμένοι τεχνικοί κανόνες, όπως οι κινητοί μέσοι όροι, μπορεί σε ορισμένες αγορές και χρονικές περιόδους να παράγουν καλύτερα αποτελέσματα από απλές στρατηγικές αναφοράς ή από τη μέση απόδοση της αγοράς.

3. Αρχιτεκτονική και τεχνολογίες εφαρμογής

Το κεφάλαιο αυτό ξεκινάει με μια σύντομη εισαγωγή στη γλώσσα Python και τις βιβλιοθήκες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Στη συνέχεια γίνεται μια παρουσίαση του Streamlit, που περιλαμβάνει και μια ιστορική αναδρομή στις τεχνολογίες των διαδικτυακών εφαρμογών για να γίνει κατανοητό πώς διαφοροποιείται το Streamlit και άλλες τεχνολογίες αυτού του τύπου από τις πιο παραδοσιακές προσεγγίσεις. Η παρουσίαση αυτή είναι αρκετά διεξοδική, καθώς το Streamlit δεν είναι απλώς ένα Python library που χρησιμοποιείται για συγκεκριμένες λειτουργίες, αλλά ένα ολοκληρωμένο framework για την ανάπτυξη διαδικτυακών εφαρμογών το οποίο καθορίζει σε μεγάλο βαθμό και την αρχιτεκτονική τους.

Το κεφάλαιο κλείνει με μια σύντομη περιγραφή της υπηρεσίας Yahoo Finance από την οποία αντλούνται οι τιμές των μετοχών και οι όγκοι συναλλαγών.

3.1. Python

3.1.1. Γενικά για την Python

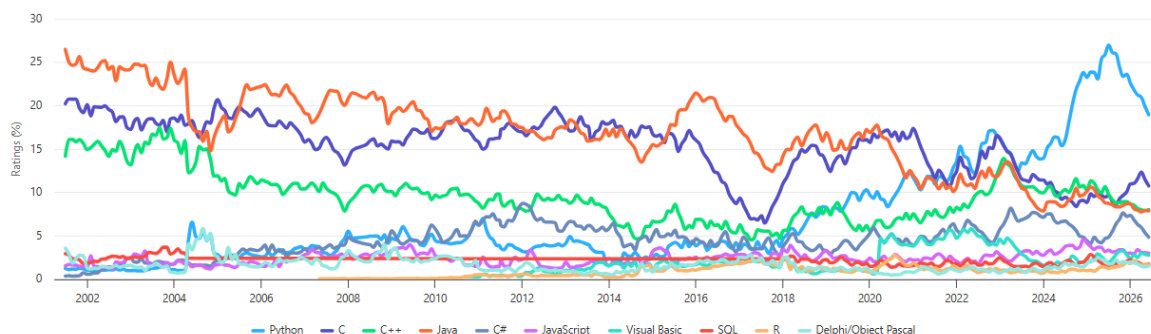
Η Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου, η οποία κυκλοφόρησε για πρώτη φορά στις αρχές της δεκαετίας του 1990. Ο σχεδιασμός και η φιλοσοφία της εστιάζουν στην απλότητα, την αναγνωσιμότητα, και την παραγωγικότητα του προγραμματιστή.

Η Python είναι interpreted (διερμηνευόμενη) και χειρίζεται τους τύπους δυναμικά (dynamically typed). Υποστηρίζει σε μεγάλο βαθμό τον αντικειμενοστραφή προγραμματισμό (object-oriented programming), ωστόσο, σε σύγκριση με γλώσσες όπως η C++ και η Java ακολουθεί πιο εύελικτη και λιγότερο φορμαλιστική προσέγγιση, καθώς δεν απαιτεί ρητές δηλώσεις τύπων για κάθε μεταβλητή, δεν χρησιμοποιεί αγκύλες για τον καθορισμό των blocks κώδικα αλλά βασίζεται στις εσοχές και τη στοίχιση, και γενικά επιτρέπει την ανάπτυξη λειτουργικού κώδικα με απλό, διαισθητικό τρόπο και μικρή συντακτική επιβάρυνση.

Επίσης, μπορεί να χρησιμοποιηθεί με διάφορα προγραμματιστικά στυλ, ανάλογα με τη φύση του προβλήματος και τους στόχους του προγραμματιστή. Για παράδειγμα, για σύντομους υπολογισμούς ή scripts μπορεί να χρησιμοποιηθεί απλός, διαδικαστικός (procedural) κώδικας, ενώ για μεγαλύτερες εφαρμογές με απαιτήσεις συντηρησιμότητας και επεκτασιμότητας μπορεί να εφαρμοστεί αντικειμενοστραφής σχεδιασμός, με κλάσεις, αφηρημένες μεθόδους, πολυμορφισμό κλπ. Παράλληλα, η Python διαθέτει και κάποια χαρακτηριστικά συναρτησιακής (functional) γλώσσας, τα οποία είναι πολύ χρήσιμα στην

επεξεργασία και το μετασχηματισμό δεδομένων, όπου συχνά εφαρμόζονται μαζικές πράξεις ή μετατροπές σε ακολουθίες, συλλογές, και πίνακες.

Τα παραπάνω χαρακτηριστικά, σε συνδυασμό με το μεγάλο αριθμό ώριμων βιβλιοθηκών που είναι διαθέσιμες για διάφορους τομείς, και ιδιαίτερα για προβλήματα ανάλυσης δεδομένων και μηχανικής μάθησης, έχουν κάνει το οικοσύστημα της Python πολύ ελκυστικό για την ανάπτυξη σύγχρονων εφαρμογών, και την Python εξαιρετικά δημοφιλή, ιδιαίτερα από το 2018 και μετά, όπως φαίνεται στην Εικόνα 3-1.



Εικόνα 3-1: TIOBE Programming Community Index. Πηγή: www.tiobe.com

Από το 2024 και μετά η Python είναι πρώτη στο δείκτη TIOBE ([2026](#)), και παραμένει πρώτη τον Ιούνιο του 2026, όπως φαίνεται στην Εικόνα 3-2. Σήμερα, η Python αποτελεί ένα από τα βασικά εργαλεία στον τομέα της ανάλυσης δεδομένων και της επιστήμης δεδομένων, και επομένως η επιλογή της για την υλοποίηση της εφαρμογής στα πλαίσια της παρούσας εργασίας είναι αυτονόητη.

Jun 2026	Jun 2025	Change	Programming Language	Ratings	Change
1	1		Python	18.96%	-6.91%
2	3	▲	C	10.77%	+1.30%
3	2	▼	C++	8.03%	-2.65%
4	4		Java	7.90%	-0.94%
5	5		C#	4.85%	+0.17%
6	6		JavaScript	3.04%	-0.17%
7	8	▲	Visual Basic	2.80%	+0.59%
8	12	▲	SQL	1.77%	+0.23%
9	14	▲	R	1.69%	+0.30%
10	9	▼	Delphi/Object Pascal	1.54%	-0.60%
11	16	▲	Scratch	1.46%	+0.27%
12	18	▲	Rust	1.26%	+0.30%
13	7	▼	Go	1.20%	-1.08%
14	15	▲	PHP	1.17%	-0.08%
15	25	▲	Swift	1.00%	+0.27%
16	11	▼	Ada	0.97%	-0.74%
17	10	▼	Fortran	0.95%	-9.91%
18	13	▼	Perl	0.92%	-0.55%
19	19		Assembly language	0.87%	-0.04%
20	17	▼	MATLAB	0.85%	-0.28%

Εικόνα 3-2: Κατάταξη γλωσσών (δείκτης TIOBE), 06/2026. Πηγή: www.tiobe.com

3.1.2. Ανάλυση και επεξεργασία δεδομένων

Η Python παρέχει δομές δεδομένων όπως λίστες (lists), λεξικά (dictionaries), πλειάδες (tuples) και σύνολα (sets), καθώς και τους μηχανισμούς slicing και comprehensions οι οποίοι διευκολύνουν την προετοιμασία και το μαζικό μετασχηματισμό δεδομένων για περαιτέρω ανάλυση, με τρόπο συντακτικά απλό και ευανάγνωστο.

Ωστόσο, για μεγάλα σύνολα δεδομένων η βασική δύναμη της Python προέρχεται από βιβλιοθήκες όπως η pandas και η NumPy, οι οποίες μεταφέρουν μεγάλο μέρος της επεξεργασίας σε ειδικές, βελτιστοποιημένες δομές και λειτουργίες. Αυτή η προσέγγιση είναι πολύ σημαντική στην ανάλυση δεδομένων που είναι οργανωμένα σε μορφή χρονοσειρών, όπου απαιτούνται πράξεις όπως μετατοπίσεις, κυλιόμενοι υπολογισμοί, υπολογισμός διαφορών, φιλτράρισμα βάσει ημερομηνίας, και προσθήκη νέων παραγόμενων στηλών.

Στην περίπτωση της παρούσας εφαρμογής, τα χρηματιστηριακά δεδομένα έχουν μορφή πίνακα, στον οποίο κάθε γραμμή αντιστοιχεί σε μία ημέρα και κάθε στήλη αντιστοιχεί είτε σε πρωτογενή δεδομένα, όπως η τιμή κλεισίματος, είτε σε αποτελέσματα υπολογισμών, όπως η τιμή κάποιου δείκτη τεχνικής ανάλυσης. Η Python, σε συνδυασμό με την pandas, επιτρέπει οι υπολογισμοί αυτοί να εκφράζονται με πολύ απλό τρόπο και να αποθηκεύονται σε νέες στήλες, όπως περιγράφεται πιο αναλυτικά στην παράγραφο 3.4.3.

3.1.3. Αντικειμενοστραφής σχεδίαση με την Python

Η Python υποστηρίζει αντικειμενοστραφή σχεδίαση και προγραμματισμό, αλλά όπως ήδη αναφέρθηκε η προσέγγισή της διαφέρει από γλώσσες όπως η Java ή η C#. Σε αυτές τις γλώσσες, οι τύποι των μεταβλητών και των επιτρεφόμενων τιμών από τις διάφορες μεθόδους είναι κατά κανόνα στατικοί και αποτελούν μέρος ενός αυστηρού “συμβολαίου” (contract). Στην Python, αντίθετα, η διαχείριση των τύπων είναι δυναμική, και τα αντικείμενα χρησιμοποιούνται με βάση τα χαρακτηριστικά και τις μεθόδους που διαθέτουν κατά την εκτέλεση του κώδικα.

Αυτό οδηγεί σε πιο ευέλικτο αλλά και λιγότερο αυστηρό μοντέλο. Η Python παρέχει μεν κλάσεις, κληρονομικότητα, και πολυμορφισμό, αλλά δεν επιβάλλει με τον ίδιο τρόπο concepts όπως private members, interfaces κλπ. Επίσης, διαθέτει “type hints” που επιτρέπουν περιγραφές των τύπων, αλλά αυτές λειτουργούν περισσότερο ως τεκμηρίωση και βοήθημα για εργαλεία στατικής ανάλυσης, όπως το Pylance ή το mypy, και όχι ως μηχανισμός υποχρεωτικού ελέγχου.

Στην παρούσα εφαρμογή, εφαρμόστηκε αντικειμενοστραφής σχεδίαση με τρόπο αρκετά κοντά στις κλασικές αντικειμενοστραφείς γλώσσες, αλλά με ορισμένες προσαρμογές. Αυτή

η προσέγγιση κρίθηκε απαραίτητη για να καλυφθεί η βασική απαίτηση της επεκτασιμότητας, καθώς και για να οργανωθεί ο κώδικας με τέτοιο τρόπο ώστε να είναι πιο συντηρήσιμος και ευανάγνωστος. Οι σχεδιαστικές επιλογές για την ανάπτυξη της εφαρμογής περιγράφονται στο Κεφάλαιο 4

3.1.4. Βιβλιοθήκες pandas και Altair

Η βασική βιβλιοθήκη για τον χειρισμό δεδομένων στην εφαρμογή είναι η **pandas**. Η pandas παρέχει δύο κύριες δομές δεδομένων: τη Series, δηλαδή μονοδιάστατη δομή με ετικέτες, και το DataFrame, δηλαδή δισδιάστατη πινακοειδή δομή με γραμμές και στήλες. Όπως αναλύεται στο *Data Structures for Statistical Computing in Python* ([McKinney, 2010](#)), η σχεδίαση της pandas στοχεύει ειδικά σε πρακτικά προβλήματα χειρισμού δεδομένων που εμφανίζονται σε περιοχές όπως η στατιστική και τα χρηματοοικονομικά.

Στην εφαρμογή, το DataFrame χρησιμοποιείται ως η κύρια μορφή αναπαράστασης των ιστορικών τιμών. Αρχικά περιέχει δεδομένα όπως ημερομηνία, τιμή κλεισίματος και όγκο συναλλαγών. Στη συνέχεια εμπλουτίζεται με νέες στήλες που αντιστοιχούν σε τεχνικούς δείκτες ή σήματα συναλλαγών. Αυτή η προσέγγιση επιτρέπει η ίδια δομή δεδομένων να χρησιμοποιείται τόσο για υπολογισμούς όσο και για οπτικοποίηση.

Για την οπτικοποίηση χρησιμοποιείται η βιβλιοθήκη **Vega-Altair**, ή απλώς Altair ([Vega-Altair documentation, 2026](#)). Στις βιβλιοθήκες αυτού του είδους, ο προγραμματιστής περιγράφει τι θέλει να απεικονίσει και πώς τα πεδία των δεδομένων αντιστοιχίζονται σε οπτικά χαρακτηριστικά, όπως άξονες, χρώματα και γραμμές. Στην εφαρμογή που αναπτύχθηκε, οι τιμές της μετοχής, ο όγκος συναλλαγών και οι υπολογισμένοι τεχνικοί δείκτες αποθηκεύονται ως στήλες στο ίδιο DataFrame. Η βιβλιοθήκη Altair επιτρέπει τη δημιουργία γραφημάτων με απευθείας χρήση αυτών των στηλών από το DataFrame, αντιστοιχίζοντας για παράδειγμα τη στήλη της ημερομηνίας στον άξονα x και τις στήλες Close, Volume, ή MA20 στον άξονα y. Με αυτόν τον τρόπο, η ίδια δομή δεδομένων που χρησιμοποιείται για τους υπολογισμούς χρησιμοποιείται και για την οπτικοποίηση των αποτελεσμάτων. Παράλληλα, η Altair παρέχει τη δυνατότητα συνδυασμού πολλαπλών γραφημάτων σε panels, που επιτρέπει την προβολή της τιμής, του όγκου και των τεχνικών δεικτών σε ξεχωριστά γραφήματα με κοινό άξονα χρόνου.

3.2. Διαδικτυακές εφαρμογές – Ιστορική αναδρομή

Ιστορικά, το Web ξεκίνησε ως σύστημα ανάκτησης, προβολής και διασύνδεσης στατικών σελίδων HTML, αλλά σταδιακά εξελίχθηκε σε πλατφόρμα εκτέλεσης σύνθετων εφαρμογών λογισμικού.

3.2.1. Server-side rendering και MVC frameworks

Το πρώτο βήμα σε αυτήν την κατεύθυνση ήταν η παραγωγή δυναμικών σελίδων HTML, που έγινε ιδιαίτερα διαδεδομένη από τα μέσα και τα τέλη της δεκαετίας του 1990, με την εμφάνιση τεχνολογιών όπως η PHP, το ASP και τα JavaServer Pages (JSP). Οι τεχνολογίες αυτές επέτρεψαν την ενσωμάτωση λογικής στην παραγωγή Web σελίδων και αποτέλεσαν τη βάση για μεγάλο αριθμό πρώιμων διαδικτυακών εφαρμογών. Κοινό χαρακτηριστικό των εφαρμογών αυτών είναι το Server-Side Rendering, δηλαδή η δημιουργία δυναμικού HTML περιεχομένου από τον server, το οποίο αποστέλλεται στο πρόγραμμα περιήγησης (browser) του χρήστη. Στο μοντέλο αυτό, ο server είναι υπεύθυνος όχι μόνο για την επιχειρησιακή λογική και την επικοινωνία με τη βάση δεδομένων, αλλά και για την παραγωγή του μεγαλύτερου μέρους του HTML/markup που καθορίζει τη δομή του UI (της διεπαφής χρήστη).

Καθώς οι διαδικτυακές εφαρμογές γίνονταν πιο σύνθετες, προέκυψε η ανάγκη για καλύτερο διαχωρισμό ευθυνών μεταξύ διαχείρισης δεδομένων, λογικής ελέγχου και παρουσίασης. Σε αυτό το πλαίσιο υιοθετήθηκαν αρχιτεκτονικά πρότυπα όπως το **Model–View–Controller (MVC)** και παραλλαγές του, που προσαρμόστηκαν για το Web. Το MVC έχει ως σκοπό να διαχωρίσει τα επιχειρησιακά δεδομένα και τη σχετική λογική (Model), από την παρουσίαση της πληροφορίας στον χρήστη (View) και τον χειρισμό των ενεργειών του χρήστη και της ροής της εφαρμογής (Controller). Στον χώρο των web εφαρμογών, η λογική αυτή εφαρμόστηκε με διάφορες μορφές, όπως το ASP.NET από τη Microsoft, το Model 2 για εφαρμογές JSP/Servlets από τη Sun, και μεταγενέστερα web frameworks όπως το Ruby on Rails που εμφανίστηκε το 2004, και το Django, το οποίο κυκλοφόρησε δημόσια το 2005. Το Rails υιοθέτησε ρητά το MVC, ενώ το Django ακολουθεί μια παραλλαγή γνωστή ως Model–Template–View (MTV).

3.2.2. Ajax και μονοσελιδικές εφαρμογές (SPA)

Από τις αρχές της δεκαετίας του 2000, οι διαδικτυακές εφαρμογές άρχισαν να συγκρίνονται με desktop εφαρμογές ως προς την ταχύτητα απόκρισης και την αμεσότητα της αλληλεπίδρασης με τον χρήστη, και η ανάγκη για καλύτερη και πιο διαδραστική εμπειρία χρήστη (User Experience, UX) έγινε ιδιαίτερα έντονη. Εφαρμογές όπως το Gmail και το Google Maps ανέδειξαν την αξία της ασύγχρονης επικοινωνίας μεταξύ client και server και της ενημέρωσης τμημάτων της σελίδας από κώδικα που εκτελείται στον browser χωρίς πλήρη ανανέωση της σελίδας. Η προσέγγιση αυτή έγινε ευρύτερα γνωστή ως **AJAX (Asynchronous JavaScript and XML)** μετά το εμβληματικό άρθρο *Ajax: A New Approach to Web Applications* ([Garrett, 2005](#)) και αποτέλεσε σημαντικό βήμα προς την εμφάνιση των **Single Page Applications (SPA)**, δηλαδή των «μονοσελιδικών» εφαρμογών.

Στην αρχιτεκτονική SPA, ο server στέλνει συνήθως μια βασική HTML σελίδα μαζί με τον απαραίτητο κώδικα JavaScript. Από εκεί και πέρα, η JavaScript αναλαμβάνει την ενημέρωση και απόδοση του UI απευθείας στον browser του χρήστη, ενώ η επικοινωνία με τον server γίνεται κυρίως μέσω APIs, συνήθως με ανταλλαγή ακατέργαστων δεδομένων σε μορφή JSON. Από το 2010 και μετά, σημειώθηκε σημαντική εξέλιξη σε αυτόν τον τομέα, με την εμφάνιση ισχυρών client-side frameworks και βιβλιοθηκών όπως η AngularJS το 2010 από την Google, το React το 2013 από τη Meta και το Vue.js το 2014. Οι υλοποιήσεις αυτού του είδους επέτρεψαν τη δημιουργία διαδικτυακών εφαρμογών με υψηλό βαθμό διαδραστικότητας και εξαιρετική εμπειρία χρήστη, στο ίδιο επίπεδο με αυτήν μιας τοπικής εφαρμογής.

Η μετάβαση από το server-side rendering στην αρχιτεκτονική SPA βελτιώνει πολλά από τα χαρακτηριστικά των διαδικτυακών εφαρμογών και ιδιαίτερα την εμπειρία χρήστη, ωστόσο αυξάνει σημαντικά την πολυπλοκότητα των υλοποιήσεων και το κόστος ανάπτυξης και συντήρησης. Η λειτουργία μιας σχεδόν ανεξάρτητης εφαρμογής στον browser του χρήστη απαιτεί γνώση JavaScript και του αντίστοιχου framework, και εισάγει μια σειρά νέων προβλημάτων σχετικά με την ασφάλεια, τη διαχείριση των εξαρτήσεων, κλπ.

Όταν υπάρχουν υψηλές απαιτήσεις για το user interface της εφαρμογής, ή σε πληροφοριακά συστήματα μεγάλης κλίμακας, ιδιαίτερα σε καταμελημένα συστήματα, όπου η ανάπτυξη των back-end APIs είναι ούτως ή άλλως απαραίτητη και οι επιχειρησιακές απαιτήσεις μπορεί να καλύπτονται με πολλαπλούς clients (π.χ. συνδυασμό από desktop και mobile applications), η πολυπλοκότητα της αρχιτεκτονικής αυτής είναι αποδεκτή και διαχειρίσιμη. Απλούστερες εφαρμογές δεν δικαιολογούν πάντα αυτό το τεχνικό βάρος.

3.2.3. Εναλλακτικές προσεγγίσεις και UI-As-Code

Το γεγονός αυτό έχει οδηγήσει τα τελευταία χρόνια σε νέες, εναλλακτικές προσεγγίσεις, που αναφέρονται συχνά ως Server-Driven ή Server-Orchestrated UI (δηλαδή διεπαφή χρήστη που οδηγείται ή ενορχηστρώνεται από τον server), ή UI-as-Code (διεπαφή χρήστη με μορφή κώδικα). Κοινό χαρακτηριστικό σε αυτές τις προσεγγίσεις είναι η περιγραφή του UI με δηλωτικό (declarative) τρόπο. Ο προγραμματιστής δεν αναπτύσσει ξεχωριστή εφαρμογή για το frontend, αντίθετα, δηλώνει αφαιρετικά στοιχεία (όπως widgets, πίνακες και γραφήματα), συχνά στην ίδια γλώσσα προγραμματισμού με το backend, και ένα εξειδικευμένο framework αναλαμβάνει αυτόματα την απόδοσή τους (rendering) στον browser, καθώς και τον συγχρονισμό της κατάστασης μεταξύ client και server σε πραγματικό χρόνο.

Η λογική του Server-Driven UI έχει χρησιμοποιηθεί σε mobile εφαρμογές από μεγάλες εταιρείες τεχνολογίας με στόχο τη δυναμική προσαρμογή τμημάτων του UI από τον server

και τον περιορισμό της ανάγκης για πλήρη δημοσίευση νέας έκδοσης της εφαρμογής για κάθε αλλαγή στο UI. Σε τέτοιες περιπτώσεις, η δυνατότητα κεντρικού ελέγχου της δομής ή του περιεχομένου του UI επιτρέπει ταχύτερες αλλαγές, πειραματισμούς και παραλλαγές, μειώνοντας την εξάρτηση από τους χρονοβόρους κύκλους έγκρισης των App Stores.

Παράλληλα, στον χώρο της επιστήμης δεδομένων αναπτύχθηκαν παρεμφερείς προσεγγίσεις που επηρέασαν τις εξελίξεις προς την ίδια κατεύθυνση για διαφορετικούς λόγους: Οι αναλυτές δεδομένων και οι ειδικοί στη μηχανική μάθηση διαθέτουν συχνά πολύ καλή γνώση Python ή R, χωρίς απαραίτητα να γνωρίζουν HTML, CSS, JavaScript ή τα σύνθετα JavaScript/TypeScript frameworks που απαιτούνται για την ανάπτυξη μονοσελιδικών εφαρμογών. Επιπλέον, οι δυνατότητες αυτών των frameworks δεν είναι πραγματικά απαραίτητες για τις περισσότερες εφαρμογές Data Analytics, καθώς το UI αυτών των εφαρμογών αποτελείται κατά κανόνα από τυποποιημένα στοιχεία (αριθμητικά πεδία, επιλογές, κουμπιά, πίνακες, γραφήματα κλπ.) και η αλληλεπίδραση με το χρήστη βασίζεται σε απλούς κύκλους αλλαγής επιλογών ή κριτηρίων και επανα-υπολογισμού. Στο πλαίσιο αυτό, δημιουργήθηκαν frameworks όπως το Shiny για την R και μεταγενέστερα το Streamlit για την Python (2019), στα οποία η βασική ιδέα είναι ότι ο προγραμματιστής περιγράφει το UI απευθείας στον κώδικα ανάλυσης, χωρίς να αναπτύσσει ξεχωριστή frontend εφαρμογή. Για παράδειγμα, στην περίπτωση του Streamlit ολόκληρη η εφαρμογή (ανάκτηση δεδομένων, αλγόριθμοι/υπολογισμοί, και διεπαφή χρήστη) γράφεται αποκλειστικά σε Python, όπως περιγράφεται πιο αναλυτικά στην παράγραφο 3.3. Με αυτόν τον τρόπο, λίγες γραμμές κώδικα μπορούν να δημιουργήσουν πλήρεις διαδικτυακές εφαρμογές.

Το βασικό πλεονέκτημα της προσέγγισης UI-As-Code είναι ότι επιτρέπει ταχύτατη ανάπτυξη πρωτοτύπων (prototyping).

Το βασικό μειονέκτημα είναι ότι αν μια εφαρμογή είναι πολύ απαιτητική ή περιλαμβάνει βαριές υπολογιστικές διαδικασίες, μπορεί να υπάρχουν δυσκολίες απόδοσης ή κλιμάκωσης. Επίσης, στις περισσότερες περιπτώσεις, ο έλεγχος στη διάταξη της οθόνης είναι σχετικά περιορισμένος, αφού αυτή καθορίζεται σε μεγάλο βαθμό από το framework.

Το Streamlit, στο οποίο βασίστηκε η ανάπτυξη της εφαρμογής, είναι από τα πιο αντιπροσωπευτικά παραδείγματα των frameworks αυτού του τύπου και περιγράφεται αναλυτικά στη συνέχεια.

3.3. Το Streamlit framework

3.3.1. Γενικά για το Streamlit

Το Streamlit είναι ένα open-source Python framework για την ανάπτυξη διαδραστικών web εφαρμογών, κυρίως εφαρμογών που βασίζονται σε δεδομένα, υπολογισμούς και οπτικοποιήσεις. Όπως αναφέρθηκε, η βασική του ιδέα είναι ότι η εφαρμογή γράφεται ως Python script, χωρίς να απαιτείται από τον προγραμματιστή να υλοποιήσει ξεχωριστό frontend με HTML, CSS και JavaScript. Οι εντολές του Streamlit, όπως `st.write`, `st.selectbox`, `st.button`, χρησιμοποιούνται μέσα στον κώδικα και παράγουν τα αντίστοιχα στοιχεία του UI, ή widgets ([Streamlit documentation, 2026](#)).

Εκτός όσων αναφέρθηκαν στο προηγούμενο κεφάλαιο, το Streamlit δημιουργήθηκε επίσης για να καλύψει ένα συχνό πρόβλημα σε εφαρμογές ανάλυσης δεδομένων, που είναι ότι ο υπολογιστικός κώδικας υπάρχει ήδη σε Python και χρειάζεται ένας απλός τρόπος ώστε ο χρήστης να αλληλεπιδρά με αυτόν μέσω browser. Αντί να αναπτυχθεί πλήρης web εφαρμογή με ξεχωριστό frontend και backend, οι σελίδες και η ροή της εφαρμογής περιγράφονται απευθείας στο Python script.

3.3.2. Αρχιτεκτονική και μοντέλο εκτέλεσης

Μια εφαρμογή Streamlit είναι εφαρμογή τύπου **client-server** (πελάτη-εξυπηρετητή) και η εκτέλεση της αρχίζει συνήθως με την εντολή:

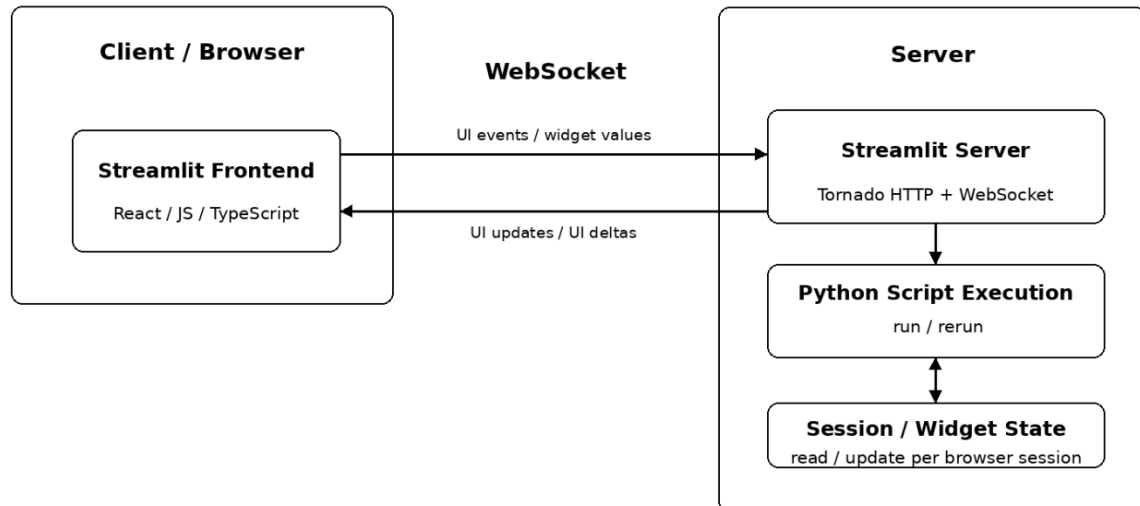
```
streamlit run app.py
```

Με την εντολή αυτή ξεκινά ο **Streamlit server**, ο οποίος εκτελεί τον κώδικα Python που βρίσκεται στο αρχείο `app.py`. Στην πλευρά του client (στον browser) δεν εκτελείται κάποιο frontend που έχει αναπτυχθεί ειδικά για τη συγκεκριμένη εφαρμογή, αλλά το τυποποιημένο **Streamlit frontend**, το οποίο είναι μια JavaScript / TypeScript εφαρμογή βασισμένη στο React framework. Η εφαρμογή αυτή, αφού φορτωθεί στον browser από το αντίστοιχο URL, εμφανίζει τα widgets, τους πίνακες, τα γραφήματα και τα υπόλοιπα στοιχεία της σελίδας, που καθορίζει το Python script που εκτελείται στον server.

Ο Streamlit server είναι βασισμένος στο Tornado web framework και η επικοινωνία του με τον client γίνεται μέσω ενός **WebSocket**, δηλαδή μιας μόνιμης αμφίδρομης σύνδεσης. Μέσω αυτής της σύνδεσης ο client στέλνει στον server τα γεγονότα που συμβαίνουν κατά την αλληλεπίδραση με τον χρήστη, όπως μια αλλαγή στην τιμή ενός πεδίου, και ο server στέλνει ενημερώσεις για το UI, τις οποίες ο client εφαρμόζει χωρίς πλήρη επαναφόρτωση της σελίδας. Ο server αποθηκεύει τις τρέχουσες τιμές και άλλες πληροφορίες κατάστασης για τα διάφορα στοιχεία του UI στο λεγόμενο **session state**, το οποίο διατηρείται για κάθε

WebSocket ξεχωριστά. Κάθε browser tab ή παράθυρο που ανοίγει την εφαρμογή δημιουργεί ένα νέο WebSocket και επομένως ένα νέο session.

Ένα διάγραμμα υψηλού επιπέδου που δείχνει την αρχιτεκτονική του Streamlit δίνεται στο Σχήμα 3-1.



Σχήμα 3-1: Λογική αρχιτεκτονική του Streamlit

Το βασικό χαρακτηριστικό του Streamlit, και αυτό που το διαφοροποιεί από άλλα frameworks του είδους, είναι ότι ο server δημιουργεί τις ενημερώσεις για τον client εκτελώντας το script γραμμικά, από πάνω προς τα κάτω, και επανεκτελώντας το από την αρχή (**rerun**) κάθε φορά που ο χρήστης αλληλεπιδρά με το UI. Σε κάθε επανεκτέλεση, η προηγούμενη κατάσταση των στοιχείων του UI ανακτάται από το session state.

Για παράδειγμα, ο παρακάτω κώδικας έχει σαν αποτέλεσμα την εμφάνιση ενός dropdown list, δηλαδή μιας πτυσσόμενης λίστας επιλογών, την ανάθεση της τρέχουσας επιλογής στη μεταβλητή `period` και την εμφάνιση ενός πληροφοριακού μηνύματος ακριβώς από κάτω:

```
import streamlit as st
...
...
period = st.selectbox("Period", ["1mo", "3mo", "6mo", "1y", "2y", "5y"], index=3)
st.write(f"Current selection: {period}")
...
```

Κατά την πρώτη εκτέλεση του script, η μεταβλητή `period` παίρνει την default τιμή από τη λίστα. Κάθε φορά που ο χρήστης αλλάζει επιλογή στη λίστα, το script επανεκτελείται από την αρχή, η μεταβλητή `period` παίρνει νέα τιμή, και το πληροφοριακό μήνυμα αλλάζει. Παρόμοια, την πρώτη φορά που εκτελείται ο παρακάτω κώδικας, η συνάρτηση `st.button()` επιστρέφει `false`, και αυτό έχει σαν αποτέλεσμα μόνο την εμφάνιση του κουμπιού:

```
import streamlit as st
...
...
if st.button("Ok"):
    st.write("Ok button clicked")
...
```

Αν ο χρήστης πατήσει το κουμπί, το script επανεκτελείται από την αρχή, αλλά αυτή τη φορά η συνάρτηση `st.button()` επιστρέφει `true`, με αποτέλεσμα να εμφανίζεται ακριβώς κάτω από το κουμπί ένα πληροφοριακό μήνυμα.

Ο κώδικας της εφαρμογής μπορεί επίσης να αποθηκεύσει αντικείμενα στο `session state`, όπως αποτελέσματα υπολογισμών ή άλλες πληροφορίες που πρέπει να διατηρούνται μεταξύ διαδοχικών εκτελέσεων του script. Για παράδειγμα, η παρακάτω συνάρτηση αποθηκεύει στο `session state` ένα αντικείμενο τύπου `CalculationResult`:

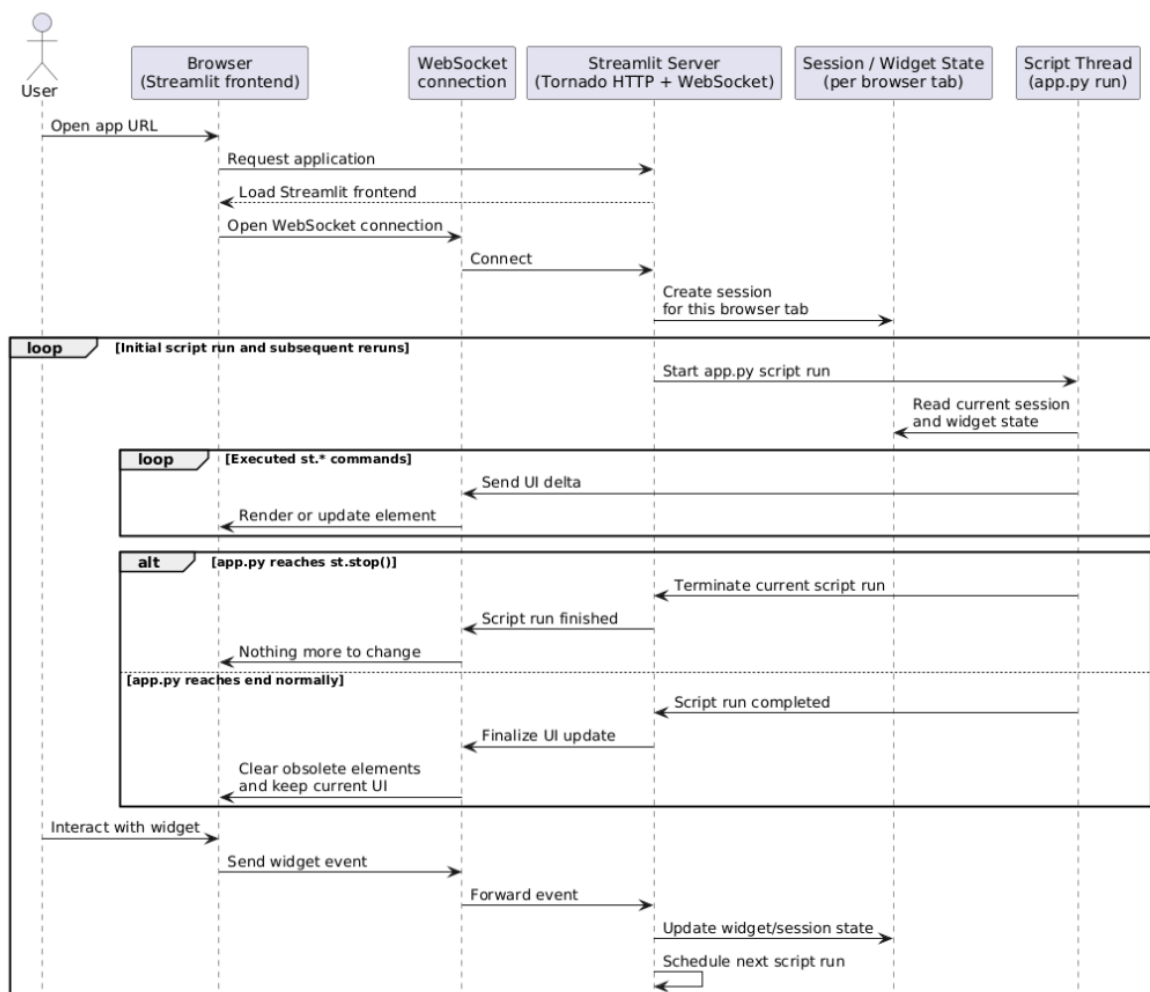
```
def set_result(result: CalculationResult) -> None:
    st.session_state["result"] = result
```

Η ανάκτηση του αντικειμένου γίνεται χρησιμοποιώντας το ίδιο κλειδί με το οποίο αποθηκεύτηκε:

```
def get_result() -> CalculationResult:
    return cast(CalculationResult, st.session_state["result"])
```

Η εκτέλεση του script μπορεί να διακοπεί προγραμματιστικά με την εντολή `st.stop()`, για παράδειγμα αν η εφαρμογή δεν πρέπει να συνεχίσει πριν δοθεί κάποια απαραίτητη είσοδος. Με τον τρόπο αυτό, μια εφαρμογή μπορεί να χτίζει το UI προοδευτικά, εμφανίζοντας επόμενα πεδία ή αποτελέσματα μόνο αφού συμπληρωθούν τα προηγούμενα.

Τα παραπάνω συνοψίζονται στο υψηλού επιπέδου `sequence diagram` που ακολουθεί (Σχήμα 3-2).



Σχήμα 3-2: Κύκλος εκτέλεσης με το Streamlit

3.3.3. Πλεονεκτήματα και περιορισμοί

Το βασικό πλεονέκτημα του Streamlit είναι ότι επιτρέπει την ανάπτυξη διαδραστικών web εφαρμογών απευθείας σε Python. Αυτό μειώνει την ανάγκη για ξεχωριστό frontend development, ειδικά όταν η εφαρμογή έχει ως κύριο αντικείμενο την επεξεργασία και παρουσίαση δεδομένων.

Επιπλέον, ο χρόνος ανάπτυξης μειώνεται μέσω των έτοιμων μηχανισμών που παρέχει το Streamlit για όλες τις βασικές ανάγκες αυτών των εφαρμογών, όπως τη δημιουργία πεδίων εισαγωγής, λιστών επιλογής, πινάκων, γραφημάτων, καθώς και την αποθήκευση δεδομένων στο state, το caching, και το deployment. Έτσι, η διαδικασία ανάπτυξης επικεντρώνεται περισσότερο στη λογική της εφαρμογής και τους υπολογισμούς, και λιγότερο σε αυτές τις βασικές υποδομές.

Σημαντικό πλεονέκτημα είναι επίσης και η δυνατότητα σταδιακής ανάπτυξης και γρήγορης δοκιμής της εφαρμογής. Η εφαρμογή εκτελείται τοπικά, και με κάθε αλλαγή στον κώδικα τα αποτελέσματα μπορούν να ελεγχθούν άμεσα στον browser. Η διαδικασία αυτή ταιριάζει πολύ σε πρωτότυπα, ερευνητικές εφαρμογές, dashboards και εσωτερικά εργαλεία.

Οι σημαντικότεροι περιορισμοί του Streamlit σχετίζονται είτε με τη δυνατότητα λεπτομερούς ελέγχου του UI, είτε με την απόδοση της εφαρμογής και τη δυνατότητα κλιμάκωσης.

Σχετικά με το UI, το Streamlit παρέχει έτοιμα widgets και αρκετές εναλλακτικές λύσεις για το layout της εφαρμογής και τους μηχανισμούς πλοήγησης, αλλά δεν δίνει τον ίδιο βαθμό ελευθερίας με μια custom εφαρμογή σε React, Angular ή Vue. Αν απαιτείται πολύ ειδική εμπειρία χρήστη, πολύπλοκη πλοήγηση ή προχωρημένο styling, η υλοποίηση με το Streamlit γίνεται πιο δύσκολη.

Σχετικά με την απόδοση, ο βασικός περιορισμός του Streamlit προκύπτει από την επανεκτέλεση του script, η οποία απαιτεί πειθαρχία στον σχεδιασμό της εφαρμογής και συγκεκριμένα μέτρα για να αποφευχθεί η επανεκτέλεση χρονοβόρων τμημάτων κώδικα σε κάθε αλληλεπίδραση με το χρήστη. Αυτό επιτυγχάνεται με τη χρήση caching καθώς και κάποιων εναλλακτικών μηχανισμών που παρέχει το Streamlit όπως οι φόρμες και τα fragments.

Σε ό,τι αφορά την κλιμάκωση, επειδή η εφαρμογή βασίζεται σε συνδέσεις WebSocket που πρέπει να παραμένουν ανοιχτές, σε εκτέλεση Python στον server ως μέρος της αλληλεπίδρασης με το χρήστη, καθώς και στο session state το οποίο διατηρείται κάθε φορά σε ένα συγκεκριμένο Streamlit server, η χρήση της εφαρμογής σε παραγωγικό περιβάλλον χρειάζεται προσοχή. Αυτό ισχύει ιδιαίτερα αν υπάρχουν αυξημένες απαιτήσεις ασφαλείας, οριζόντιας κλιμάκωσης, logging, ή monitoring.

Συνολικά, το Streamlit είναι κατάλληλο όταν ζητούμενο είναι η γρήγορη ανάπτυξη μιας διαδραστικής εφαρμογής με Python, αλλά όχι όταν η κύρια απαίτηση είναι πολύ σύνθετο frontend, αυστηρός έλεγχος της εμπειρίας χρήστη ή πλήρης enterprise αρχιτεκτονική μεγάλης κλίμακας.

3.3.4. Χρήση του Streamlit στην εφαρμογή

Για μια πειραματική εφαρμογή ανάλυσης μετοχών σε εκπαιδευτικό περιβάλλον, το Setramlit αποτελεί μία εξαιρετική επιλογή.

Τα μειονεκτήματα που αναφέρθηκαν σχετικά με τον έλεγχο του UI, ούτως ή άλλως δεν είναι σημαντικά για εφαρμογές Data Analytics, ενώ τα σχετικά με την κλιμάκωση δεν έχουν εφαρμογή στα πλαίσια της εργασίας.

Σχετικά με την απόδοση, έχουν ληφθεί μέτρα στην εφαρμογή ώστε να αποφεύγεται η ανάκτηση δεδομένων σε κάθε rerun (caching) καθώς και η επανεκτέλεση χρονοβόρων υπολογισμών σε κάθε αλληλεπίδραση με το χρήστη, ιδιαίτερα στην υλοποίηση των στρατηγικών που μπορούν να εκτελεστούν για πολλές μετοχές ταυτόχρονα. Ο τρόπος υλοποίησης περιγράφεται πιο αναλυτικά στο Κεφάλαιο 4.

3.4. Yahoo Finance

3.4.1. Επιλογή της υπηρεσίας Yahoo Finance

Σε οποιαδήποτε εφαρμογή ανάλυσης μετοχών, απαιτείται η χρήση μιας υπηρεσίας παροχής των αντίστοιχων δεδομένων, όπως τιμών κλεισίματος και όγκων. Για τις ανάγκες της παρούσας εργασίας επιλέχθηκε το Yahoo Finance, το οποίο χρησιμοποιείται μέσω της βιβλιοθήκης της Python **yfinance**.

Το Yahoo Finance (<https://finance.yahoo.com>) αποτελεί μία από τις πιο γνωστές και αξιόπιστες δημόσια προσβάσιμες πηγές χρηματιστηριακών δεδομένων, καθώς και βασικών χρηματοοικονομικών δεδομένων και θεμελιωδών μεγεθών, και η χρήση της μέσω της βιβλιοθήκης yfinance είναι εξαιρετικά διαδεδομένη στο οικοσύστημα της Python, ιδιαίτερα σε πειραματικές εφαρμογές, εκπαιδευτικά περιβάλλοντα, και ερευνητικές εργασίες χρηματοοικονομικής ανάλυσης. Είναι χαρακτηριστικό ότι στο GitHub repository της βιβλιοθήκης ([yfinance GitHub repository, 2026](#)) υπάρχουν πάνω από 24k “GitHub stars” και 3,5k “forks”, που δείχνει ότι το yfinance είναι εξαιρετικά δημοφιλές.

Το Yahoo Finance καλύπτει πλήρως τις ανάγκες της παρούσας εργασίας καθώς παρέχει ιστορικά στοιχεία τιμών για μεγάλο πλήθος χρηματιστηριακών αγορών, περιλαμβανομένων των αμερικανικών αγορών NASDAQ και NYSE, του αγγλικού LSE, καθώς και του Χρηματιστηρίου Αθηνών (ASE), και δεν απαιτεί σύνθετους μηχανισμούς πιστοποίησης ή κλειδιά πρόσβασης.

3.4.2. Η βιβλιοθήκη yfinance

Η βιβλιοθήκη yfinance λειτουργεί ως «περίβλημα» (wrapper) της υπηρεσίας και επιτρέπει την ανάκτηση χρηματοοικονομικών δεδομένων με απλό τρόπο μέσω Python.

Η βασική έννοια της βιβλιοθήκης είναι το αντικείμενο **Ticker**, το οποίο αντιστοιχεί σε ένα χρηματιστηριακό σύμβολο (μία μετοχή). Μέσω του αντικειμένου αυτού είναι δυνατή η ανάκτηση ιστορικών δεδομένων για διαφορετικά χρονικά διαστήματα, καθώς και πρόσθετων στοιχείων για τη μετοχή. Στην παρούσα εργασία χρησιμοποιήθηκε κυρίως η μέθοδος `history()`, μέσω της οποίας ανακτώνται ιστορικά δεδομένα σε μορφή **pandas DataFrame**. Ενδεικτικό παράδειγμα χρήσης:

```
ticker = yf.Ticker("AAPL")  
hist = ticker.history(period="1y")
```

Η βιβλιοθήκη μπορεί να ανακτήσει δεδομένα για διάφορες περιόδους, όπως τον τελευταίο 1 μήνα, 6 μήνες, 1 έτος, 5 έτη, αλλά και για συγκεκριμένα χρονικά διαστήματα.

Για τις μετοχές του Χρηματιστηρίου Αθηνών χρησιμοποιείται η κατάληξη .AT, σύμφωνα με τη σύμβαση ονοματοδοσίας του Yahoo Finance. Για παράδειγμα, η μετοχή του ΟΠΑΠ αναφέρεται ως OPAP.AT.

3.4.3. Δομές δεδομένων

Τα δεδομένα που επιστρέφονται από τη βιβλιοθήκη yfinance οργανώνονται σε αντικείμενα τύπου pandas DataFrame. Τα DataFrames αποτελούν βασική δομή δεδομένων της βιβλιοθήκης pandas και χρησιμοποιούνται ευρέως σε εφαρμογές ανάλυσης δεδομένων και χρονοσειρών. Ένα τυπικό DataFrame ιστορικών τιμών περιλαμβάνει πεδία όπως ημερομηνία, τιμή ανοίγματος και κλεισίματος, μέγιστη και ελάχιστη τιμή ημέρας, όγκος συναλλαγών κλπ.

Η χρήση των DataFrames διευκολύνει σημαντικά την υλοποίηση δεικτών τεχνικής ανάλυσης, καθώς παρέχει ενσωματωμένες δυνατότητες επεξεργασίας χρονοσειρών και υπολογισμού στατιστικών μεγεθών. Για παράδειγμα, ο υπολογισμός ενός κινητού μέσου όρου 20 ημερών, που απαιτεί τον ορισμό ενός κυλίσμενου «παραθύρου» 20 ημερών και τον υπολογισμό του μέσου όρου της τιμής κλεισίματος για κάθε τέτοιο παράθυρο, μπορεί να πραγματοποιηθεί με μία μόνο εντολή:

```
df["MA20"] = df["Close"].rolling(window=20).mean()
```

Αντίστοιχα, οι εκθετικοί κινητοί μέσοι όροι υλοποιούνται μέσω της συνάρτησης ewm() της βιβλιοθήκης pandas.

Η συγκεκριμένη προσέγγιση αποτέλεσε βασικό στοιχείο της αρχιτεκτονικής της εφαρμογής καθώς επέτρεψε την υλοποίηση ενός γενικευμένου και επεκτάσιμου μηχανισμού υπολογισμού τεχνικών δεικτών. Περισσότερες λεπτομέρειες για την χρήση των DataFrames στην εφαρμογή δίνονται στην παράγραφο 4.3.

3.4.4. Ενσωμάτωση στην εφαρμογή

Στην εφαρμογή υλοποιήθηκε ξεχωριστό επίπεδο δεδομένων (Data Layer), υπεύθυνο για την ανάκτηση και διαχείριση των χρηματιστηριακών δεδομένων από το Yahoo Finance. Ο διαχωρισμός αυτός συνέβαλε στη μείωση της εξάρτησης μεταξύ του επιπέδου δεδομένων, της λογικής υπολογισμού των δεικτών (business logic) και διεπαφής χρήστη (user interface).

Για βελτίωση της απόδοσης χρησιμοποιήθηκε ο μηχανισμός caching του Streamlit (@st.cache_data), ώστε να αποφεύγεται η επαναλαμβανόμενη ανάκτηση των ίδιων δεδομένων.

Η εφαρμογή υποστηρίζει επιλογή μετοχής και χρονικού διαστήματος ανάλυσης από το χρήστη, ανακτά τα δεδομένα από το Yahoo Finance και στη συνέχεια τα τροφοδοτεί στο υποσύστημα υπολογισμού δεικτών, παραγωγής σημάτων αγοράς και πώλησης, και οπτικοποίησης δεδομένων.

3.4.5. Περιορισμοί της υπηρεσίας

Παρά τα σημαντικά της πλεονεκτήματα, η υπηρεσία Yahoo Finance έχει και ορισμένους περιορισμούς, με σημαντικότερο το γεγονός ότι δεν αποτελεί επίσημο πάροχο δεδομένων μέσω API, ούτε δεσμεύεται από καθορισμένα επίπεδα εξυπηρέτησης. Συνεπώς, υπάρχει πάντα η πιθανότητα να παρουσιαστούν καθυστερήσεις ενημέρωσης ή περιορισμοί στη διαθεσιμότητα των δεδομένων.

Επιπλέον, η βιβλιοθήκη yfinance χρησιμοποιεί μη τεκμηριωμένα APIs του Yahoo Finance καθώς και τεχνικές “scrapping”, δηλαδή ανάλυση του περιεχομένου που εμφανίζεται στις σελίδες του σχετικού ιστότοπου (<https://finance.yahoo.com/>). Αυτό σημαίνει ότι σε περίπτωση δραστικών αλλαγών από την πλευρά του Yahoo Finance ενδέχεται να παρουσιάσει προβλήματα.

Οι περιορισμοί αυτοί δεν επηρεάζουν ουσιαστικά τους στόχους αυτής της εργασίας, καθώς η εφαρμογή ανακτά μόνο πολύ βασικά στοιχεία για πειραματισμό και όχι για επενδυτικές αποφάσεις, και η πιθανότητα ολικής μη διαθεσιμότητας αυτών των δεδομένων θεωρείται αμελητέα.

4. Σχεδιασμός και υλοποίηση

Στο κεφάλαιο αυτό παρουσιάζονται οι σχεδιαστικές αρχές και τα patterns που έχουν χρησιμοποιηθεί για να παραχθεί κατανοητός και εύκολα συντηρήσιμος κώδικας, και να καλυφθεί η βασική, μη λειτουργική απαίτηση της επεκτασιμότητας.

Επίσης, περιγράφεται ο τρόπος ανάκτησης δεδομένων, παρουσίασης του UI και αλληλεπίδρασης με το χρήστη, και ο τρόπος υλοποίησης των δεικτών, των σημάτων αγοράς και πώλησης, και των στρατηγικών συναλλαγών.

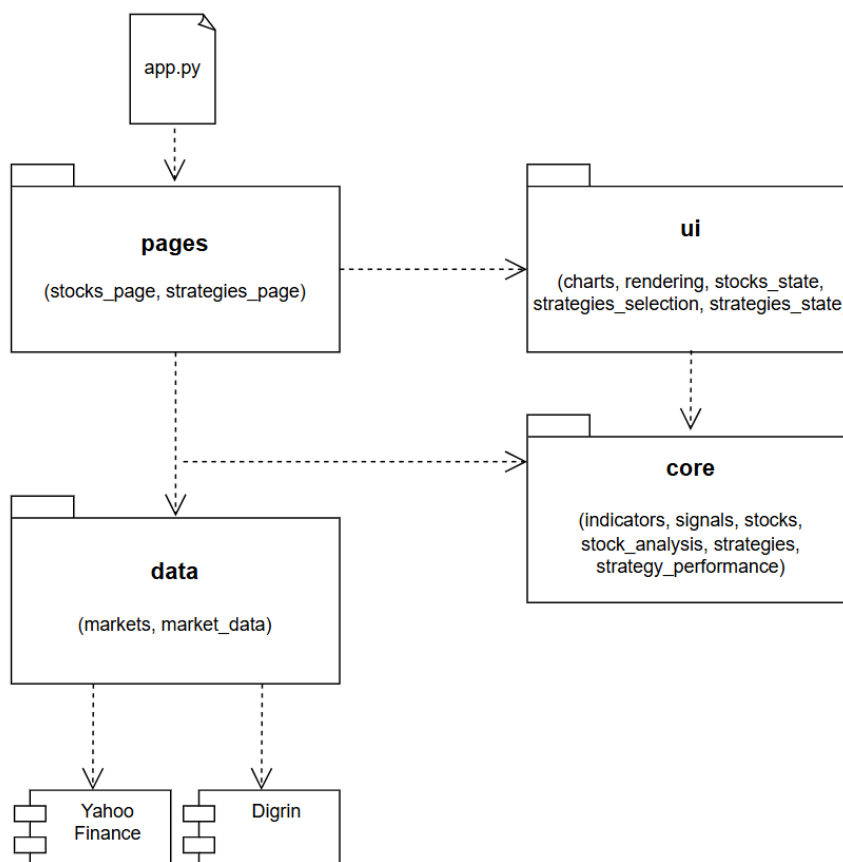
4.1. Οργάνωση κώδικα

Η εφαρμογή οργανώνεται σε πακέτα (packages), τα οποία αντιστοιχούν στους βασικούς φακέλους του πηγαίου κώδικα. Σύμφωνα με τις συμβάσεις της Python, οι φάκελοι αυτοί περιέχουν το ειδικό αρχείο “__init__.py”, το οποίο επιτρέπει στην Python να τους αναγνωρίζει ως ενότητες κώδικα που μπορούν να εισαχθούν με “import”. Τα επιμέρους αρχεία “.py” που περιέχονται σε αυτά τα packages αποτελούν τα Python modules της εφαρμογής.

Η Python δεν επιβάλλει αυστηρό αρχιτεκτονικό μοντέλο για τη διάρθρωση μιας εφαρμογής. Αντίστοιχα, το Streamlit δεν επιβάλλει κάποιο πρότυπο όπως το MVC, και σε κάποιο βαθμό ευνοεί την ανάμιξη διαφορετικών αρμοδιοτήτων στον ίδιο κώδικα, λόγω του τρόπου με τον οποίο υλοποιεί τη διεπαφή με το χρήστη (το UI). Για τον λόγο αυτό, στην παρούσα εφαρμογή υιοθετήθηκε μια απλή δομή πακέτων, ώστε οι βασικές αρμοδιότητες να είναι στοιχειωδώς διαχωρισμένες.

Η οργάνωση αυτή παρουσιάζεται στο διάγραμμα πακέτων που ακολουθεί (Σχήμα 4-1). Το διάγραμμα δείχνει το **κεντρικό script “app.py”** και τα packages της εφαρμογής, μαζί με τα modules που περιλαμβάνονται σε καθένα από αυτά, καθώς και τις εξαρτήσεις μεταξύ των packages. Τα βέλη δηλώνουν ότι modules ενός package εισάγουν και χρησιμοποιούν συναρτήσεις, κλάσεις, ή σταθερές, που ορίζονται σε άλλα packages. Στο διάγραμμα εμφανίζονται επίσης οι εξωτερικές πηγές δεδομένων Yahoo Finance και Diggrin, οι οποίες δεν αποτελούν packages της εφαρμογής, αλλά συστήματα από τα οποία αντλούνται δεδομένα.

Το “app.py” αποτελεί το σημείο εισόδου της εφαρμογής. Σε αυτό ορίζεται η αρχική παραμετροποίηση της εφαρμογής, η γενική διάταξη της αρχικής σελίδας, η πλοήγηση προς τις επιμέρους σελίδες και μια λίστα επιλογής αγοράς (χρηματιστηρίου) η οποία παραμένει πάντα διαθέσιμη. Το “app.py” διαμορφώνει το βασικό περιβάλλον της εφαρμογής και δίνει στον χρήστη πρόσβαση στις σελίδες του πακέτου pages.



Σχήμα 4-1: Διάγραμμα πακέτων και εξωτερικών πηγών δεδομένων

Το πακέτο **pages** περιλαμβάνει τα scripts των λειτουργικών σελίδων της εφαρμογής. Η σελίδα “stocks_page.py” υποστηρίζει την ανάλυση μεμονωμένης μετοχής, ενώ η σελίδα “strategies_page.py” υποστηρίζει την αξιολόγηση στρατηγικών συναλλαγών για μία ή περισσότερες μετοχές. Τα modules αυτά έχουν κυρίως ρόλο συντονισμού, ελέγχοντας τη ροή της αλληλεπίδρασης με το χρήστη: διαβάζουν τις επιλογές από το UI, καλούν τις λειτουργίες φόρτωσης δεδομένων, ενεργοποιούν τους υπολογισμούς και αναθέτουν την εμφάνιση των αποτελεσμάτων στα modules του επόμενου πακέτου.

Το πακέτο **ui** περιλαμβάνει τον κώδικα που σχετίζεται με την παρουσίαση και την κατάσταση της διεπαφής χρήστη. Σε αυτό ανήκουν τα modules που δημιουργούν τα διαγράμματα, αποδίδουν κοινά στοιχεία της διεπαφής, διαχειρίζονται τις επιλογές στρατηγικών και αποθηκεύουν την κατάσταση των σελίδων μέσω του session_state του Streamlit. Με αυτόν τον τρόπο, ο κώδικας των σελίδων παραμένει πιο συγκεντρωμένος στη ροή της εφαρμογής και δεν επιβαρύνεται με όλες τις λεπτομέρειες της παρουσίασης.

Το πακέτο **core** περιλαμβάνει την κεντρική υπολογιστική λογική της εφαρμογής. Σε αυτό ορίζονται οι τεχνικοί δείκτες, τα σήματα αγοράς και πώλησης, οι στρατηγικές συναλλαγών και οι συναρτήσεις υπολογισμού της απόδοσης των στρατηγικών.

Το πακέτο **data** περιλαμβάνει τον κώδικα που σχετίζεται με την ανάκτηση και προετοιμασία δεδομένων από τις εξωτερικές πηγές, καθώς και τις ρυθμίσεις των υποστηριζόμενων αγορών (χρηματιστηρίων). Μέσω αυτού του πακέτου η εφαρμογή επικοινωνεί με το Yahoo Finance, που περιγράφεται στην παράγραφο 3.4, από το οποίο ανακτά ιστορικά δεδομένα και το Digrin (<https://www.digrin.com/>) από το οποίο ανακτά λίστες διαθέσιμων μετοχών ανά χρηματιστήριο.

Η δομή αυτή επιτρέπει στην εφαρμογή να παραμένει συμβατή με το απλό μοντέλο ανάπτυξης του Streamlit, ενώ ταυτόχρονα διατηρεί έναν σαφή διαχωρισμό αρμοδιοτήτων μεταξύ των διαφόρων τμημάτων του κώδικα.

4.2. Ανάκτηση δεδομένων

Η ανάκτηση δεδομένων στην εφαρμογή γίνεται μέσω συναρτήσεων που ορίζονται στο πακέτο data. Ο τρόπος ανάκτησης και διαχείρισης των δεδομένων περιγράφεται παρακάτω συνοπτικά.

4.2.1. Διαθέσιμες μετοχές

Η υπηρεσία Yahoo Finance και επομένως η βιβλιοθήκη yfinance δεν παρέχουν με απλό και ενιαίο τρόπο κατάλογο των διαθέσιμων μετοχών ανά χρηματιστήριο. Υπάρχουν διάφορες λύσεις για αυτό το κοινό πρόβλημα, μεταξύ των οποίων η χρήση στατικών λιστών ή βιβλιοθηκών που αναλύουν τις σελίδες του Yahoo Finance ή άλλων sites (scrapping). Εδώ επιλέχθηκε το site <https://www.digrin.com/>, το οποίο εμφανίζει λίστες μετοχών ανά χρηματιστήριο. Στην εφαρμογή έχει οριστεί ένα URL template της μορφής https://www.digrin.com/stocks/list/exchanges/{exchange_code} όπου το {exchange_code} αντικαθίσταται από τον κωδικό του χρηματιστηρίου, όπως ath, nyse, nasdaq ή lse. Η ανάκτηση γίνεται στέλνοντας ένα HTTP request στη συγκεκριμένη διεύθυνση και διαβάζοντας το HTML περιεχόμενο της σελίδας για να εντοπιστεί ο πίνακας που περιέχει τις μετοχές. Από κάθε γραμμή του πίνακα εξάγεται το σύμβολο της μετοχής και η επωνυμία της εταιρείας.

Οι λίστες που ανακτώνται από το Digrin δεν είναι πάντοτε ενημερωμένες, μπορεί δηλαδή μια μετοχή να εμφανίζεται στο Digrin αλλά να έχει αποσυρθεί ή να έχει αλλάξει σύμβολο. Για τον λόγο αυτό, μετά την αρχική ανάκτηση, κάθε σύμβολο ελέγχεται ως προς τη διαθεσιμότητα ιστορικών τιμών στο Yahoo Finance. Μετοχές για τις οποίες το Yahoo Finance δεν επιστρέφει δεδομένα εξαιρούνται από τις λίστες επιλογής της εφαρμογής.

4.2.2. Ιστορικά δεδομένα και θεμελιώδη

Για την ανάκτηση ιστορικών δεδομένων χρησιμοποιείται η βιβλιοθήκη `yfinance` και συγκεκριμένα η μέθοδος `history()` του αντικειμένου `Ticker`. Από το `DataFrame` που επιστρέφει η μέθοδος, χρησιμοποιείται η τιμή κλεισίματος (`Close`) και ο όγκος συναλλαγών (`Volume`).

Η ανάκτηση των δεδομένων γίνεται με έναρξη χρονικού διαστήματος παλαιότερη από αυτή που έχει επιλέξει ο χρήστης, ώστε να είναι δυνατός ο υπολογισμός δεικτών και για τις πρώτες ημέρες του επιλεγμένου χρονικού διαστήματος. Αυτές οι επιπλέον τιμές αφαιρούνται (σε άλλα πακέτα της εφαρμογής) πριν την δημιουργία των γραφημάτων.

Εκτός από τα ιστορικά δεδομένα, η εφαρμογή αντλεί και ορισμένα βασικά στοιχεία για τη λειτουργία αναζήτησης και ταξινόμησης μετοχών. Για τον σκοπό αυτό χρησιμοποιείται το λεξικό `info` που είναι διαθέσιμο στο `Ticker`. Από το λεξικό αυτό ανακτώνται εγγραφές όπως `longName` και `shortName` για την ονομασία της εταιρείας, `marketCap` για την κεφαλαιοποίηση κλπ. Οι εγγραφές αυτές δεν είναι ομοιόμορφα συμπληρωμένες για όλα τα χρηματιστήρια και τις μετοχές, για αυτό και η εφαρμογή χρησιμοποιεί σε κάποιες περιπτώσεις λογική “fallback”, δηλαδή ελέγχει περισσότερες πιθανές τιμές και επιστρέφει την πρώτη διαθέσιμη.

Τα δεδομένα που χρησιμοποιεί η εφαρμογή από το λεξικό `Ticker.info` δίνονται στον παρακάτω πίνακα.

Πηγή	Τύπος	Περιγραφή	Παράδειγμα
<code>Ticker.info["longName"]</code>	string	Πλήρης ονομασία εταιρείας	"Aegean Airlines S.A."
<code>Ticker.info["shortName"]</code>	string	Εναλλακτική / σύντομη ονομασία εταιρείας	"Aegean Airlines"
<code>Ticker.info["sector"]</code>	string	Κλάδος	"Industrials"
<code>Ticker.info["sectorDisp"]</code>	string	Εναλλακτικό πεδίο εμφάνισης κλάδου	"Industrials"
<code>Ticker.info["industry"]</code>	string	Υποκλάδος	"Airlines"
<code>Ticker.info["industryDisp"]</code>	string	Εναλλακτικό πεδίο εμφάνισης υποκλάδου	"Airlines"
<code>Ticker.info["marketCap"]</code>	integer	Κεφαλαιοποίηση	1080000000
<code>Ticker.info["dividendYield"]</code>	float	Μερισματική απόδοση	0.0617
<code>Ticker.info["trailingPE"]</code>	float	Ιστορικός δείκτης P/E	9.85
<code>Ticker.info["forwardPE"]</code>	float	Εκτιμώμενος μελλοντικός δείκτης P/E	8.7
<code>Ticker.info["trailingEps"]</code>	float	Κέρδη ανά μετοχή	1.25
<code>Ticker.info["averageVolume"]</code>	integer	Μέσος όγκος συναλλαγών	245000
<code>Ticker.info["previousClose"]</code>	float	Προηγούμενη τιμή κλεισίματος	12.2
<code>Ticker.info["regularMarketPreviousClose"]</code>	float	Εναλλακτική προηγούμενη τιμή κλεισίματος	12.2
<code>Ticker.info["currentPrice"]</code>	float	Τρέχουσα / πρόσφατη τιμή	12.34

Πηγή	Τύπος	Περιγραφή	Παράδειγμα
<code>Ticker.info["regularMarketPrice"]</code>	float	Εναλλακτική τρέχουσα / πρόσφατη τιμή	12.34
<code>Ticker.info["fiftyTwoWeekLow"]</code>	float	Χαμηλό 52 εβδομάδων	9.8
<code>Ticker.info["fiftyTwoWeekHigh"]</code>	float	Υψηλό 52 εβδομάδων	13.1
<code>Ticker.info["52WeekChange"]</code>	float	Απόδοση 52 εβδομάδων	0.184
<code>Ticker.info["currency"]</code>	string	Νόμισμα	"EUR"

Πίνακας 4-1: Δεδομένα που χρησιμοποιεί η εφαρμογή από το λεξικό Ticker.info

Για λόγους απόδοσης, η εφαρμογή χρησιμοποιεί caching σε περισσότερα από ένα επίπεδα. Στις σελίδες της εφαρμογής, συναρτήσεις όπως η ανάκτηση συμβόλων, η ανάκτηση ιστορικών δεδομένων και η δημιουργία του DataFrame για τη λειτουργία αναζήτησης έχουν δηλωθεί με `@st.cache_data`. Έτσι, όταν ο χρήστης επιλέγει την ίδια αγορά, μετοχή ή περίοδο, η εφαρμογή δεν εκτελεί ξανά το ίδιο εξωτερικό αίτημα, αλλά χρησιμοποιεί το αποθηκευμένο αποτέλεσμα. Η χρήση caching είναι ιδιαίτερα σημαντική για τη λίστα συμβόλων και για τα δεδομένα αναζήτησης, επειδή αυτά μπορεί να απαιτούν πολλές επιμέρους κλήσεις στις εξωτερικές πηγές.

Ανακεφαλαιώνοντας, η στρατηγική ανάκτησης δεδομένων βασίζεται σε πρακτικό συνδυασμό πηγών. Η εφαρμογή λαμβάνει υπόψη τους περιορισμούς και τις ασυνέπειες των δημόσια διαθέσιμων πηγών και εφαρμόζει ελέγχους και λογική fallback. Με τον τρόπο αυτό επιτυγχάνεται ικανοποιητική λειτουργικότητα για τους σκοπούς της εργασίας, χωρίς να απαιτείται χρήση εμπορικής υπηρεσίας για τα χρηματιστηριακά δεδομένα.

4.3. Χρήση των DataFrames

Στην εφαρμογή, η διαχείριση των αρχικών δεδομένων από το Yahoo Finance και των αποτελεσμάτων των υπολογισμών, γίνεται μέσω αντικειμένων τύπου DataFrame που είναι η κεντρική δομή που χρησιμοποιεί η βιβλιοθήκη pandas για την επεξεργασία δεδομένων σε μορφή πίνακα.

Η χρήση των DataFrames είναι μία από τις βασικότερες αρχές σχεδιασμού της εφαρμογής και διατρέχει όλες τις κλάσεις που σχετίζονται με ιστορικά δεδομένα, υπολογισμούς δεικτών και εκτέλεση στρατηγικών, καθώς ένα DataFrame περνάει διαδοχικά από instances αυτών των κλάσεων και εμπλουτίζεται με νέες στήλες.

Στην εφαρμογή κάθε γραμμή του DataFrame αντιστοιχεί σε μία ημερομηνία, δηλαδή σε μία συνεδρίαση, και οι πρώτες στήλες περιλαμβάνουν τα ιστορικά δεδομένα που ανακτώνται από το Yahoo Finance (ημερομηνία, τιμή κλεισίματος και όγκος συναλλαγών). Στη συνέχεια το DataFrame εμπλουτίζεται με υπολογισμούς τεχνικών δεικτών. Επειδή οι δείκτες μπορεί να είναι πάνω από ένας και εξαρτώνται από τις επιλογές του χρήστη, οι σχετικές στήλες προστίθενται δυναμικά, με ονομασίες που είναι μοναδικές για κάθε δείκτη

και συνδυασμό παραμέτρων. Εφόσον στη συνέχεια εκτελεστεί κάποια στρατηγική συναλλαγών (back testing), τα αρχικά αποτελέσματά της αποθηκεύονται επίσης σε νέες στήλες στο DataFrame. Τα αποτελέσματα αυτά περιλαμβάνουν τα πρωτογενή σήματα αγοράς και πώλησης, τα οποία είναι διαθέσιμα για περαιτέρω επεξεργασία, όπως εφαρμογή φίλτρων και υπολογισμό απόδοσης. Ακολουθεί σχετικό παράδειγμα (Πίνακας 4-2).

Ιστορικά στοιχεία			Τιμές δεικτών			Αρχικά αποτελέσματα στρατηγικής			
Date	Close	Volume	MA20	MA50	...	BUY_condition	SELL_condition	BUY_signal	SELL_signal
02/03/2026	€ 10.10	123,000	9.92	10.05	...	FALSE	TRUE	FALSE	FALSE
03/03/2026	€ 10.35	118,500	10.01	10.04	...	FALSE	TRUE	FALSE	FALSE
04/03/2026	€ 10.50	132,000	10.06	10.05	...	TRUE	FALSE	TRUE	FALSE
05/03/2026	€ 10.62	141,000	10.13	10.06	...	TRUE	FALSE	FALSE	FALSE
06/03/2026	€ 10.48	127,500	10.17	10.07	...	TRUE	FALSE	FALSE	FALSE
09/03/2026	€ 10.20	150,000	10.08	10.09	...	FALSE	TRUE	FALSE	TRUE
...	...	...	...	...	...	...	...	...	...

Πίνακας 4-2: Παράδειγμα χρήσης DataFrame στην εφαρμογή

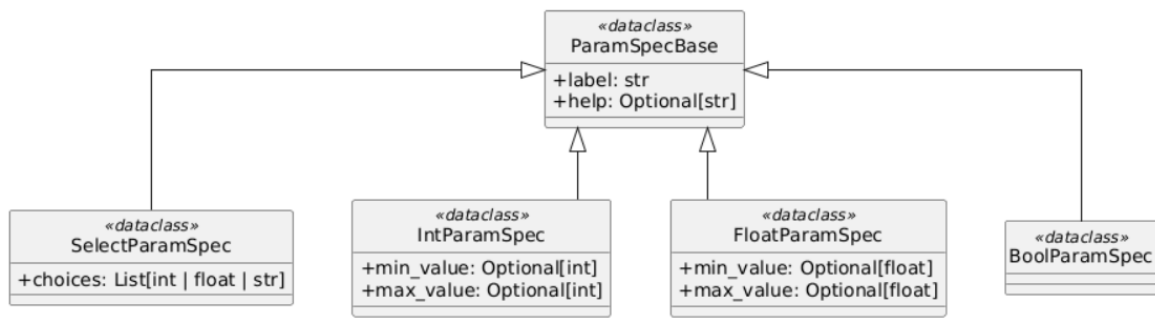
Σημειώνεται ότι τα ίδια δεδομένα χρησιμοποιούνται και για οπτικοποίηση, αφού πρώτα αντιγραφούν σε νέο DataFrame και προστεθούν στήλες σχετικές με την παρουσίαση των γραφημάτων. Ενδεικτικά, για τα γραφήματα γίνεται αλλαγή της τιμής που αντιστοιχεί στον οριζόντιο άξονα ώστε να μην εμφανίζονται κενά όπου δεν υπάρχουν διαθέσιμα στοιχεία (αργίες, σαββατοκύριακα), καθώς και αλλαγή της ονομασίας των στηλών, προκειμένου να χρησιμοποιηθούν από τον γενικό μηχανισμό δημιουργίας γραφημάτων και τελικά τη βιβλιοθήκη Altair.

Τα DataFrames προσφέρουν πολύ μεγάλη ευελιξία και παίζουν σημαντικό ρόλο στο οικοσύστημα της Python, ωστόσο διαφέρουν από πιο παραδοσιακές προσεγγίσεις, όπως τα typed datasets όπου η δομή των δεδομένων ορίζεται πιο αυστηρά και μπορεί να ελεγχθεί στατικά πριν την εκτέλεση του κώδικα. Για να περιοριστεί ο κίνδυνος σφαλμάτων κατά την εκτέλεση, οι ονομασίες των στηλών που χρησιμοποιούνται στην εφαρμογή ορίζονται κεντρικά ως σταθερές ή ως ιδιότητες των κλάσεων.

4.4. Δείκτες

Οι τεχνικοί δείκτες υλοποιούνται στο πακέτο core, ως υποκλάσεις της αφηρημένης κλάσης Indicator, και χρησιμοποιούνται από την υπόλοιπη εφαρμογή όλοι με τον ίδιο τρόπο. Κάθε δείκτης υλοποιεί τον δικό του αλγόριθμο υπολογισμού και παρέχει metadata που καθορίζουν πώς παραμετροποιείται και πώς απεικονίζεται.

Για την παραμετροποίηση, χρησιμοποιείται ένα μικρό μοντέλο περιγραφής παραμέτρων, με βάση την κλάση ParamSpecBase και τις υποκλάσεις της (Σχήμα 4-2). Ο σκοπός του μοντέλου αυτού είναι κάθε δείκτης να υλοποιεί τις παραμέτρους του με δηλωτικό (declarative) τρόπο, χωρίς να απαιτείται ειδικός κώδικας για το χειρισμό τους.



Σχήμα 4-2: Διάγραμμα κλάσεων παραμετροποίησης

Για παράδειγμα, με τον παρακάτω κώδικα η κλάση `MovingAverage` που υλοποιεί τον κινητό μέσο όρο ορίζει ότι χρησιμοποιεί μία αριθμητική παράμετρο με όνομα `period` και “ετικέτα” `Period`, η οποία ορίζεται μέσω μιας λίστας επιλογών με πιθανές τιμές 5, 10, 20, 50, 200:

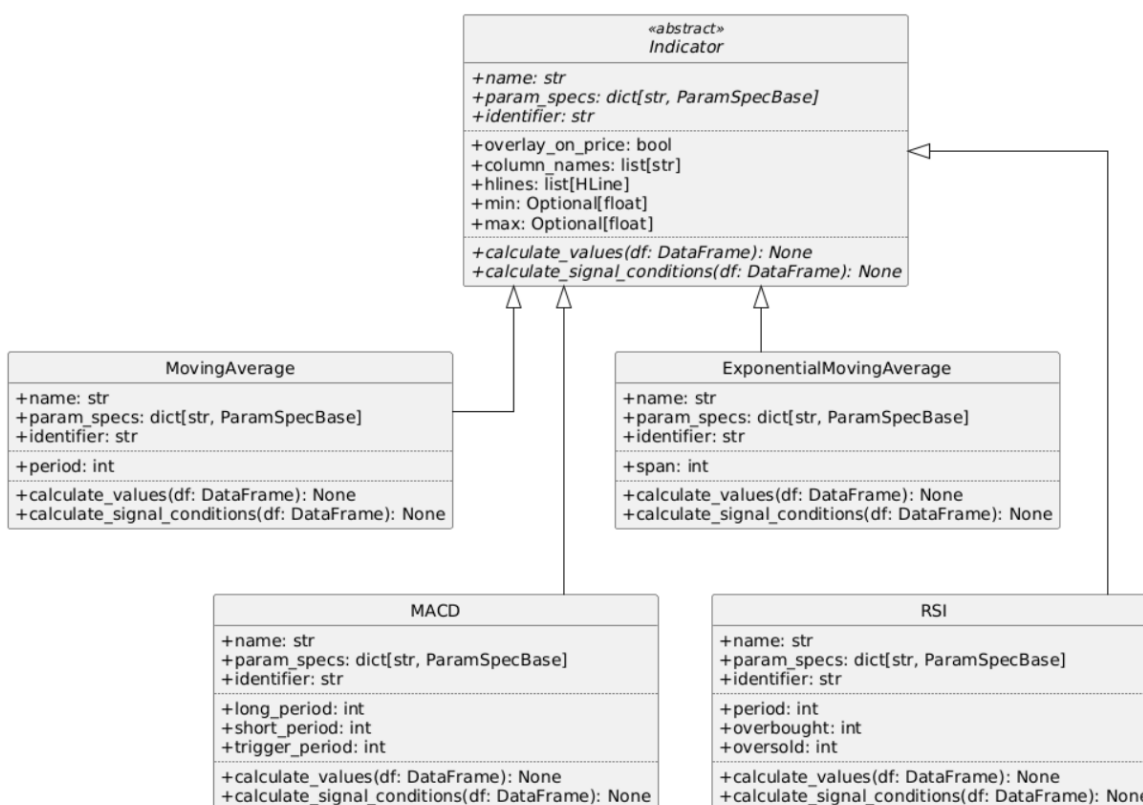
```

param_specs = {
    "period": SelectParamSpec(label="Period", choices=[5, 10, 20, 50, 200])
}

```

Για τον τρόπο απεικόνισης ενός δείκτη χρησιμοποιούνται συγκεκριμένες ιδιότητες που έχουν οριστεί στην βασική κλάση `Indicator` και ελέγχουν διάφορα χαρακτηριστικά, μεταξύ άλλων το αν ο δείκτης προβάλλεται μαζί με την τιμή, όπως ο κινητός μέσος, ή σε ξεχωριστό γράφημα, όπως ο `MACD`, αν στο γράφημα πρέπει να εμφανίζονται οριζόντιες γραμμές ως επίπεδα αναφοράς, όπως τα επίπεδα 30 και 70 στο δείκτη `RSI`, καθώς και προαιρετικά όρια για τον κάθετο άξονα. Στα πλαίσια της τεχνικής ανάλυσης τα στοιχεία αυτά δεν είναι επιλογές απεικόνισης που μεταβάλλονται ελεύθερα, αλλά χαρακτηριστικά των ίδιων των δεικτών, και η υλοποίησή τους απευθείας στις αντίστοιχες κλάσεις είναι κατάλληλη σχεδιαστική επιλογή για την εργασία.

Στο διάγραμμα κλάσεων που ακολουθεί (Σχήμα 4-3) εμφανίζεται η αφηρημένη κλάση `Indicator` με τις βασικές της ιδιότητες και μεθόδους, καθώς και ενδεικτικές υλοποιήσεις συγκεκριμένων δεικτών. Το διάγραμμα αναδεικνύει ότι οι δείκτες έχουν κοινό `interface`, αλλά διαφοροποιούνται ως προς τις παραμέτρους, τον τρόπο υπολογισμού και τον τρόπο απεικόνισής τους.



Σχήμα 4-3: Διάγραμμα κλάσεων δεικτών

Ακολουθούν οι ελάχιστες ιδιότητες και μέθοδοι που απαιτείται να οριστούν για την υλοποίηση ενός δείκτη (Πίνακας 4-3). Με βάση τα στοιχεία αυτά, και αφού ο δείκτης προστεθεί σε έναν κατάλογο (registry), οι γενικοί μηχανισμοί της εφαρμογής καλύπτουν αυτόματα την παραμετροποίηση του δείκτη από το χρήστη, την προετοιμασία του DataFrame με τις τιμές της μετοχής, την εκτέλεση του υπολογισμού, και την προβολή του γραφήματος.

Ιδιότητα/Μέθοδος	Περιγραφή
name	Η γενική ονομασία του δείκτη (π.χ. “Moving average”)
identifier	Μοναδικό αναγνωριστικό για το συγκεκριμένο instance του δείκτη (π.χ. “MA20” για Moving average 20 ημερών)
param_specs	Dictionary που καθορίζει τις απαιτούμενες παραμέτρους και τον τρόπο εισαγωγής τους.
calculate_values()	Υπολογισμός του δείκτη σε νέες στήλες του DataFrame
calculate_signal_conditions()	Υπολογισμός συνθηκών αγοράς – πώλησης σε νέες στήλες του DataFrame, για χρήση από τις “προκαθορισμένες στρατηγικές” (βλ. 4.6)

Πίνακας 4-3: Βασικές ιδιότητες και μέθοδοι δεικτών

Στην Εικόνα 4-1 δίνεται ενδεικτικά η πλήρης υλοποίηση του κινητού μέσου όρου.

```
class MovingAverage(Indicator):
    name = "Moving Average"

    period: int

    param_specs = {
        "period": SelectParamSpec(label="Period", choices=[5, 10, 20, 50, 200])
    }

    def __init__(self, period: int = 20):
        self.period = period

    @property
    @override
    def identifier(self) -> str:
        return f"MA{self.period}"

    @override
    def calculate_values(self, df: pd.DataFrame) -> None:
        df[self.identifier] = df["Close"].rolling(window=self.period).mean()

    @override
    def calculate_signal_conditions(self, df: pd.DataFrame) -> None:
        df[SignalColumns.BUY_CONDITION] = df["Close"] > df[self.identifier]
        df[SignalColumns.SELL_CONDITION] = df["Close"] < df[self.identifier]
```

Εικόνα 4-1: Ο κώδικας του δείκτη Moving Average

4.5. Στρατηγικές συναλλαγών

Οι στρατηγικές συναλλαγών υλοποιούνται στο πακέτο `core` μέσω της αφηρημένης κλάσης `Strategy`, η οποία ορίζει ένα κοινό interface για όλες τις στρατηγικές.

Η κεντρική σχεδιαστική επιλογή που έχει γίνει για τις στρατηγικές και έχει μοντελοποιηθεί στην κλάση `Strategy` είναι ότι οι στρατηγικές βασίζονται κατά κανόνα σε “συνθήκες αγοράς – πώλησης”, δηλαδή σε λογικές τιμές (`true` – `false`) για κάθε ημερήσια τιμή της μετοχής, που δείχνουν αν ικανοποιούνται τα κριτήρια της στρατηγικής. Για παράδειγμα, σε μια απλή στρατηγική βασισμένη σε έναν κινητό μέσο, η “συνθήκη αγοράς” είναι αληθής για κάθε ημερομηνία στην οποία η τιμή κλεισίματος είναι μεγαλύτερη από τον κινητό μέσο. Οι συνθήκες αγοράς και πώλησης παρέχονται από τις υποκλάσεις, ενώ η βασική, αφηρημένη κλάση αναλαμβάνει:

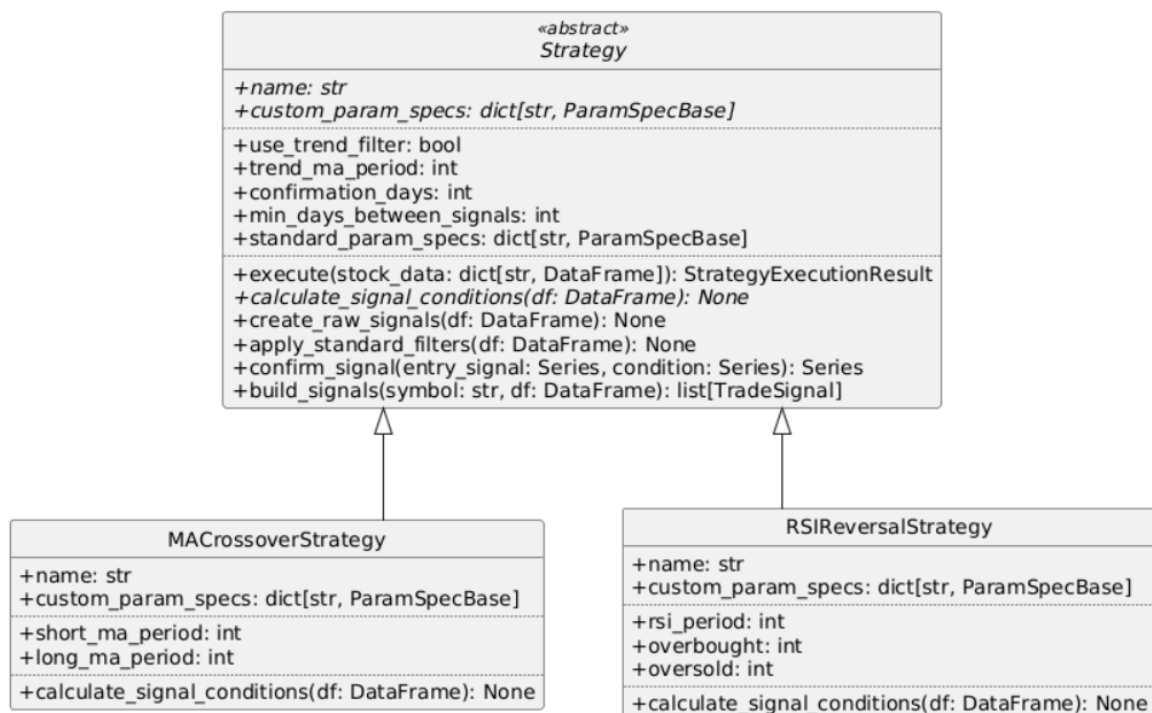
- Τον υπολογισμό σημάτων αγοράς και πώλησης, στα σημεία όπου οι παραπάνω συνθήκες αλλάζουν, όπως κατά την ανοδική ή καθοδική διάσπαση του κινητού μέσου όρου από την τιμή.
- Την εφαρμογή προαιρετικών φίλτρων στα σήματα αγοράς και πώλησης, όπως επιβεβαίωση μίας ή περισσότερων ημερών ή επιβεβαίωση ανοδικής τάσης πριν ένα σήμα αγοράς θεωρηθεί έγκυρο.
- Την εκτέλεση της στρατηγικής για πολλές μετοχές μαζί και το συνδυασμό των αποτελεσμάτων σε ένα επιστρεφόμενο αντικείμενο για περαιτέρω επεξεργασία.

Με αναφορά στο κλασικό έργο *Design Patterns: Elements of Reusable Object-Oriented Software* ([Gamma et al., 1994](#)), ο σχεδιασμός αυτών των κλάσεων είναι αρκετά κοντά στο

πρότυπο (pattern) **Template Method**, στο οποίο μία αφηρημένη κλάση ορίζει τα βήματα ενός αλγόριθμου και αφήνει την υλοποίηση ενός ή περισσότερων βημάτων στις υποκλάσεις της, μέσω ενός abstract method. Εδώ, το βήμα που υλοποιεί κάθε υποκλάση είναι ο υπολογισμός των συνθηκών αγοράς – πώλησης, και η υλοποίηση γίνεται μέσω της μεθόδου `calculate_signal_conditions()`, η οποία έχει οριστεί στην κλάση `Strategy` ως abstract. Ταυτόχρονα, υπάρχει η δυνατότητα να γίνουν override και άλλες μέθοδοι της βασικής κλάσης αν αυτό είναι απαραίτητο, όπως για την υλοποίηση στρατηγικών που απαιτούν διαφοροποίηση από το βασικό μοντέλο των συνθηκών αγοράς – πώλησης που περιγράφηκε.

Για την παραμετροποίηση των στρατηγικών ακολουθείται η ίδια προσέγγιση με τους δείκτες, με βάση την κλάση `ParamSpecBase` και τις υποκλάσεις της (βλ. 4.4). Η αφηρημένη κλάση `Strategy` ορίζει τις προκαθορισμένες παραμέτρους εκτέλεσης που δεν εξαρτώνται από τον τύπο της στρατηγικής, και οι υποκλάσεις τυχόν πρόσθετες παραμέτρους.

Για καλύτερη κατανόηση όσων αναφέρθηκαν παραπάνω, στο Σχήμα 4-4 δίνεται ένα απλοποιημένο διάγραμμα κλάσεων, που περιλαμβάνει δύο ενδεικτικές υλοποιήσεις στρατηγικών. Επισημαίνεται ότι οι κλάσεις αυτές χρησιμοποιούν εσωτερικά και τεχνικούς δείκτες, οι οποίοι δεν εμφανίζονται στο διάγραμμα. Το πλήρες διάγραμμα για όλες τις κλάσεις που εμπλέκονται στην υλοποίηση στρατηγικών δίνεται στην επόμενη παράγραφο.



Σχήμα 4-4: Απλοποιημένο διάγραμμα κλάσεων στρατηγικών

Οι ελάχιστες ιδιότητες και μέθοδοι που απαιτείται να οριστούν κατά την υλοποίηση μιας στρατηγικής δίνονται παρακάτω (Πίνακας 4-4). Με βάση τα στοιχεία αυτά, οι γενικοί μηχανισμοί της εφαρμογής καλύπτουν αυτόματα την παραμετροποίηση της στρατηγικής από το χρήστη, την επιλογή μετοχών, την προετοιμασία του DataFrame με τις τιμές τους, την εκτέλεση της στρατηγικής, και την προβολή των αποτελεσμάτων.

Ιδιότητα/Μέθοδος	Περιγραφή
name	Η γενική ονομασία της στρατηγικής (π.χ. "Moving Average Crossover")
custom_param_specs	Dictionary που ορίζει πρόσθετες παραμέτρους εκτέλεσης και τον τρόπο εισαγωγής τους από το χρήστη.
calculate_signal_conditions()	Υπολογισμός συνθηκών αγοράς – πώλησης σε νέες στήλες του Dataframe

Πίνακας 4-4: Βασικές ιδιότητες και μέθοδοι στρατηγικών

Στην Εικόνα 4-2 δίνεται ενδεικτικά η πλήρης υλοποίηση της στρατηγικής Moving Average Crossover, που βασίζεται στην ανοδική ή καθοδική διάσπαση ενός μακροπρόθεσμου από έναν βραχυπρόθεσμο κινητό μέσο.

```
class MACrossoverStrategy(Strategy):
    name = "Moving Average Crossover"

    custom_param_specs = {
        "short_ma_period": replace(MovingAverage.param_specs["period"], label="Short MA Period"),
        "long_ma_period": replace(MovingAverage.param_specs["period"], label="Long MA Period"),
    }

    def __init__(self, short_ma_period: int = 20, long_ma_period: int = 50):
        super().__init__()
        self._short_ma = MovingAverage(period=short_ma_period)
        self._long_ma = MovingAverage(period=long_ma_period)

    @property
    def short_ma_period(self) -> int:
        return self._short_ma.period

    @short_ma_period.setter
    def short_ma_period(self, value: int):
        self._short_ma.period = value

    @property
    def long_ma_period(self) -> int:
        return self._long_ma.period

    @long_ma_period.setter
    def long_ma_period(self, value: int):
        self._long_ma.period = value

    @override
    def calculate_signal_conditions(self, df: pd.DataFrame) -> None:
        self._short_ma.calculate_values(df)
        self._long_ma.calculate_values(df)

        short_col = self._short_ma.identifier
        long_col = self._long_ma.identifier

        df[SignalColumns.BUY_CONDITION] = df[short_col] > df[long_col]
        df[SignalColumns.SELL_CONDITION] = df[short_col] < df[long_col]
```

Εικόνα 4-2: Ο κώδικας της στρατηγικής Moving Average Crossover

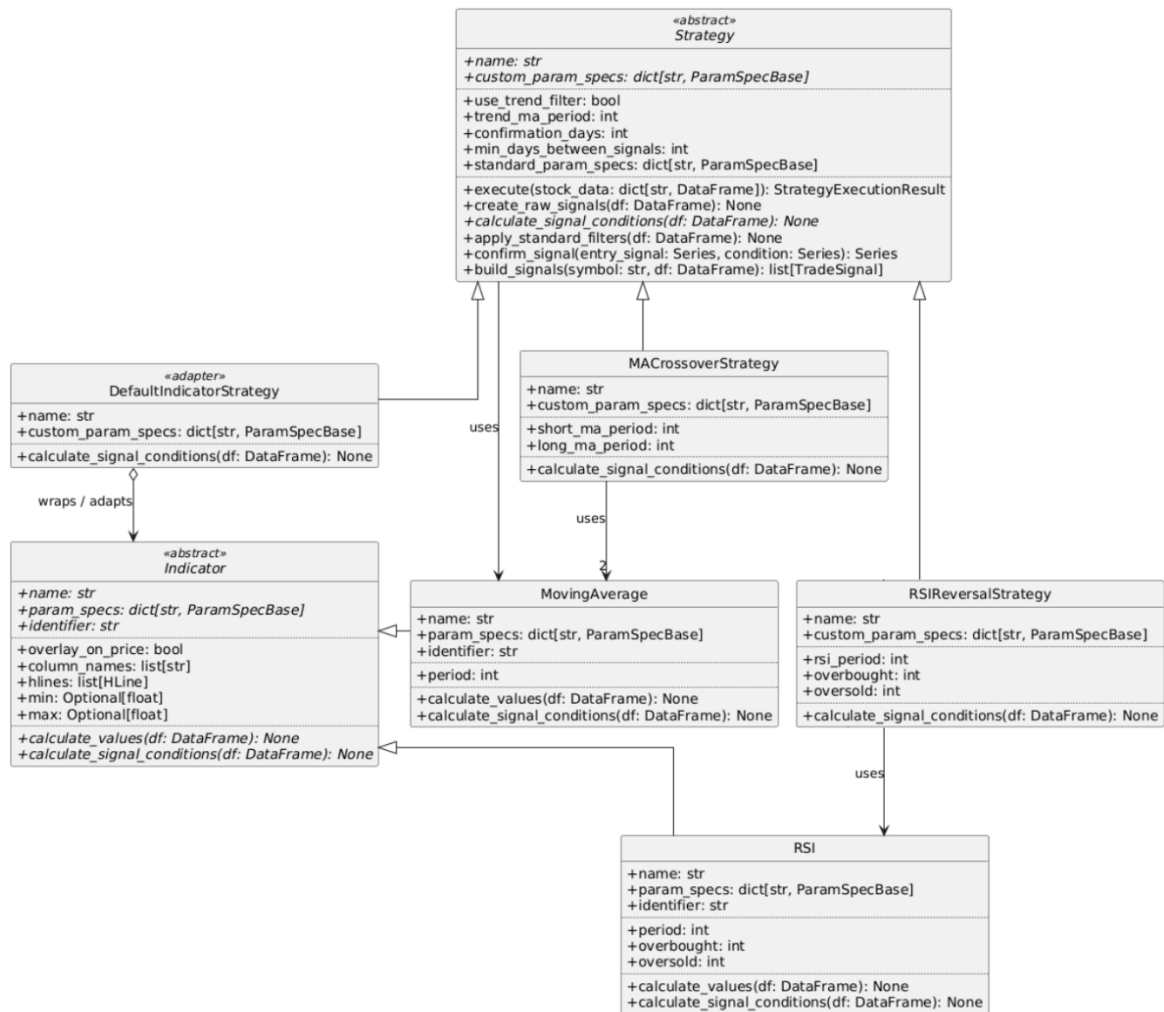
4.6. Προκαθορισμένες στρατηγικές

Για αρκετούς τεχνικούς δείκτες, η συνήθης ερμηνεία τους αντιστοιχεί και σε μία απλή στρατηγική συναλλαγών. Σε έναν κινητό μέσο όρο, για παράδειγμα, η ανοδική διάσπαση του δείκτη από την τιμή της μετοχής μπορεί να θεωρηθεί ότι παρέχει ένα πιθανό σήμα αγοράς. Στο πλαίσιο της εργασίας, αυτές οι απλές στρατηγικές ονομάζονται προκαθορισμένες (default) και η εφαρμογή παρέχει τα μέσα για την υλοποίησή τους με ελάχιστο κώδικα στον αντίστοιχο δείκτη. Για να επιτευχθεί αυτό, ο σχεδιασμός έχει επεκταθεί ως εξής:

- Στην κλάση `Indicator` έχει προστεθεί μια επιπλέον αφηρημένη μέθοδος, η `calculate_signal_conditions()`, για τον υπολογισμό των συνθηκών αγοράς – πώλησης, παρόμοια με την αντίστοιχη μέθοδο της κλάσης `Strategy`.
- Έχει δημιουργηθεί η νέα κλάση `DefaultIndicatorStrategy`, η οποία ενσωματώνει οποιονδήποτε δείκτη και υλοποιεί αυτόματα μια απλή στρατηγική με βάση αυτόν τον δείκτη.

Στο σημείο αυτό, ο σχεδιασμός των κλάσεων είναι αρκετά κοντά στο **Adapter** pattern ([Gamma et al., 1994](#)), γνωστό και ως `Wrapper`, στο οποίο μία κλάση “προσαρμόζει” ένα υπάρχον αντικείμενο σε ένα διαφορετικό `interface`, ώστε να μπορεί να χρησιμοποιηθεί από κώδικα που απαιτεί αυτό το `interface`. Η κλάση που παίζει εδώ το ρόλο του Adapter είναι η `DefaultIndicatorStrategy`, η οποία προσαρμόζει αντικείμενα τύπου `Indicator` στο `interface` της κλάσης `Strategy`. Αυτό το επιτυγχάνει εκθέτοντας όλες τις παραμέτρους του δείκτη ως παραμέτρους της στρατηγικής, και αναθέτοντας στον δείκτη τον υπολογισμό των συνθηκών αγοράς – πώλησης καλώντας τη μέθοδο `calculate_signal_conditions()`. Η υλοποίηση αυτής της μεθόδου σε κάθε δείκτη απαιτεί απλώς την προσθήκη δύο νέων στηλών στο `DataFrame` με τιμές `true` ή `false`, ανάλογα με το αν ικανοποιείται το βασικό κριτήριο του δείκτη, για παράδειγμα, αν η τιμή κλεισίματος βρίσκεται πάνω ή κάτω από τον κινητό μέσο όρο.

Ακολουθεί το πλήρες διάγραμμα για όλες τις κλάσεις που εμπλέκονται στην υλοποίηση στρατηγικών (Σχήμα 4-5).



Σχήμα 4-5: Διάγραμμα κλάσεων στρατηγικών

Όπως αναφέρθηκε ήδη, η `DefaultIndicatorStrategy` παρέχει μια απλή προκαθορισμένη στρατηγική για οποιονδήποτε δείκτη, ενώ οι άλλες δύο στρατηγικές (`MACrossoverStrategy`, `RSISReversalStrategy`) είναι παραδείγματα πιο εξειδικευμένων υλοποιήσεων.

Όπως φαίνεται και στο διάγραμμα, η `MACrossoverStrategy` χρησιμοποιεί δύο κινητούς μέσους (έναν βραχυπρόθεσμο και έναν μακροπρόθεσμο) δηλαδή δύο instances της κλάσης `MovingAverage`. Αντίστοιχα, η κλάση `RSISReversalStrategy` χρησιμοποιεί ένα instance της κλάσης `RSI`.

Τέλος, η κλάση `Strategy` (και επομένως όλες οι στρατηγικές) χρησιμοποιεί έναν κινητό μέσο όρο αν έχει ζητηθεί επιβεβαίωση τάσης (`use_trend_filter = true`). Στην περίπτωση αυτή, τα σήματα αγοράς αγνοούνται αν η τιμή της μετοχής είναι κάτω από τον επιλεγμένο κινητό μέσο.

4.7. Διαχείριση κατάστασης

Όπως περιγράφηκε αναλυτικά στην παράγραφο 3.3.2, το Streamlit διατηρεί στο session state την κατάσταση όλων των widgets που εμφανίζονται στο UI, αλλά η εφαρμογή αποθηκεύει στο session state και πρόσθετες τιμές ή αντικείμενα. Αυτό είναι απαραίτητο αφενός για να ελέγχεται η ροή της εφαρμογής και αφετέρου για να αποφεύγεται η επανεκτέλεση κώδικα για υπολογισμούς σε κάθε rerun του κεντρικού script.

Κατά τη χρήση του session state πρέπει να λαμβάνεται υπόψη ότι οι επιλογές του χρήστη έχουν αλληλεξαρτήσεις. Για παράδειγμα, όταν ο χρήστης αλλάζει χρηματιστήριο ή επιλέγει άλλη μετοχή, ορισμένα δεδομένα που έχουν αποθηκευτεί στο state παύουν να είναι έγκυρα και πρέπει να καθαριστούν. Για αυτό το λόγο, αλλά και για καλύτερη αναγνωσιμότητα, ο κώδικας διαχείρισης του state δεν είναι διάσπαρτος στις σελίδες αλλά έχει οργανωθεί κεντρικά, σε modules του πακέτου state. Με τον τρόπο αυτό, κάθε σελίδα χρησιμοποιεί απλές συναρτήσεις για την αρχικοποίηση, ανάγνωση και μεταβολή του state, ενώ ο κώδικας διαχείρισης αναλαμβάνει τις λεπτομέρειες σχετικά με τα κλειδιά που χρησιμοποιούνται, τους κανόνες αλλαγής χρηματιστηρίου, μετοχών, δεικτών, ή στρατηγικής, καθώς και την εκκαθάριση προηγούμενων αποτελεσμάτων.

Ο Πίνακας 4-5 συνοψίζει τα βασικά αντικείμενα και τις τιμές που διατηρούνται από την εφαρμογή στο session state.

Κλειδί	Τύπος	Εύρος	Περιγραφή
stocks_market	str	Global	Κωδικός επιλεγμένου χρηματιστηρίου
stocks_page_symbol	str None	Stocks	Επιλεγμένη μετοχή
stocks_page_indicators	list[Indicator]	Stocks	Αντικείμενα τύπου Indicator με τις τρέχουσες παραμέτρους τους
stocks_page_trade_signals	list None	Stocks	Λίστα σημάτων αγοράς - πώλησης εφόσον έχει ζητηθεί εκτέλεση της προκαθορισμένης στρατηγικής για κάποιον δείκτη
stocks_page_strategy_indicator_idx	int None	Stocks	Θέση στη λίστα stocks_page_indicators του δείκτη από τον οποίο έχουν παραχθεί τα σήματα αγοράς - πώλησης
strategies_page_selected_symbols	list[str]	Strategies	Επιλεγμένες μετοχές
strategies_page_strategy	Strategy None	Strategies	Επιλεγμένη στρατηγική
strategies_page_result	StrategyExecutionResult None	Strategies	Αποτελέσματα back testing ανά μετοχή και συνολικά
screeners_page_selected_symbols	list[str]	Screeners	Επιλεγμένες μετοχές

Πίνακας 4-5: Τιμές και αντικείμενα που διατηρούνται στο session state

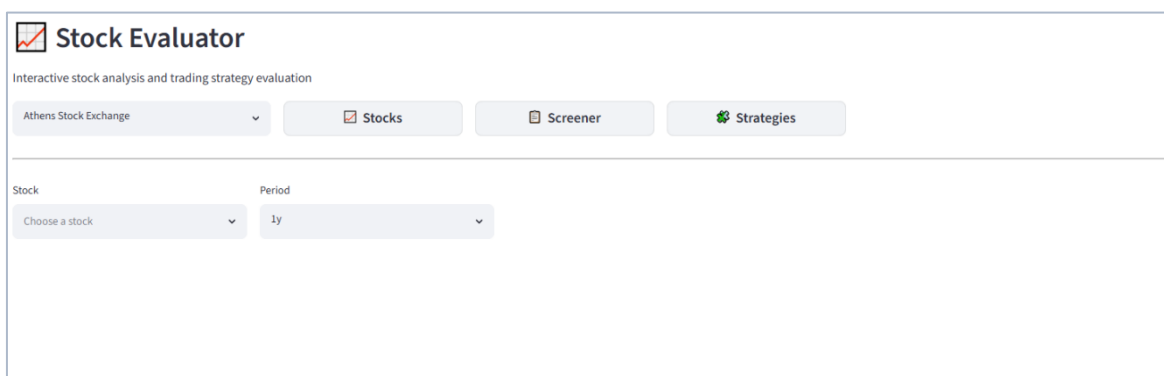
Συνολικά, η διαχείριση κατάστασης αποτελεί βασικό στοιχείο της εφαρμογής, επειδή γεφυρώνει το απλό μοντέλο εκτέλεσης του Streamlit με την ανάγκη για έλεγχο της ροής της εφαρμογής και των εξαρτήσεων μεταξύ επιλογών. Η υλοποίηση σε ξεχωριστά modules διατηρεί καθαρότερο τον κώδικα των σελίδων και αυξάνει τη συντηρησιμότητα της εφαρμογής.

5. Αποτελέσματα και αξιολόγηση

5.1. Παραδείγματα χρήσης

Με την είσοδο στην εφαρμογή, επιλέγεται αυτόματα η αγορά “Athens Stock Exchange”, δηλαδή το Χρηματιστήριο Αθηνών, και ο χρήστης κατευθύνεται στη σελίδα Stocks, όπου μπορεί να επιλέξει μετοχή και περίοδο ανάλυσης.

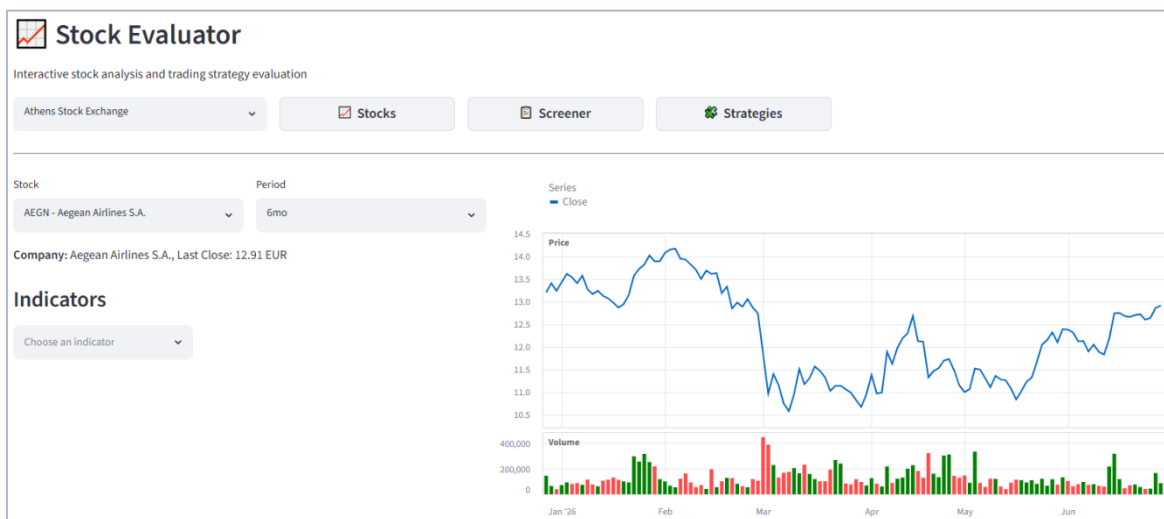
Η λίστα επιλογής αγοράς, που παραμένει πάντα διαθέσιμη, περιλαμβάνει επίσης το Χρηματιστήριο του Λονδίνου (London Stock Exchange) καθώς και τα αμερικανικά χρηματιστήρια NYSE και NASDAQ. Εφόσον επιλεγεί άλλη αγορά, η εφαρμογή ανανεώνει τη λίστα των διαθέσιμων μετοχών και επιστρέφει στην αρχική σελίδα (Εικόνα 5-1).



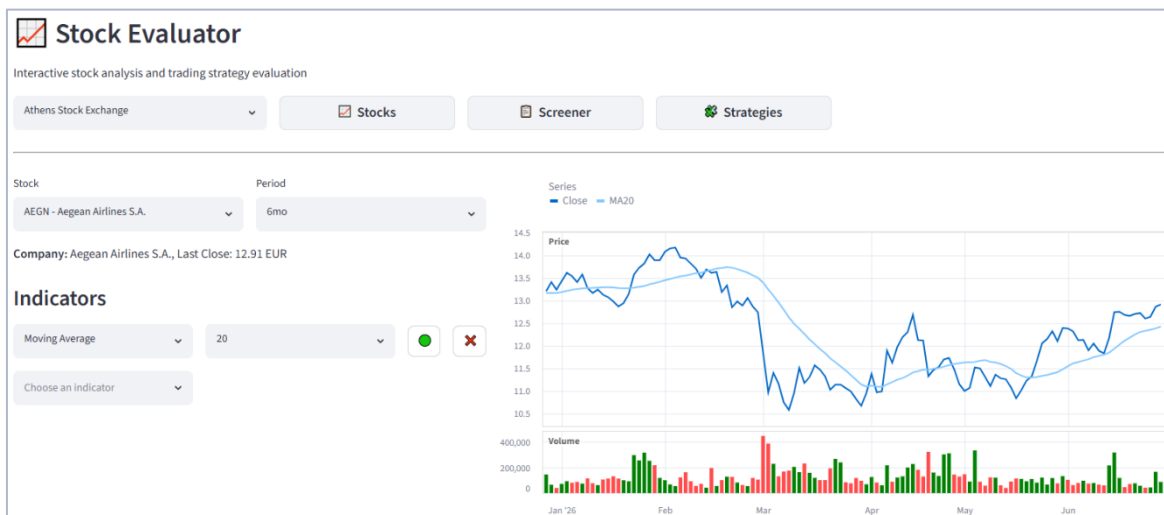
Εικόνα 5-1: Stock Evaluator – Αρχική σελίδα

Επιλέγοντας μετοχή, εμφανίζονται τα διαγράμματα τιμής και όγκου για την επιλεγμένη περίοδο. Σύμφωνα με την κοινή πρακτική σε εφαρμογές αυτού του είδους, οι όγκοι συναλλαγών εμφανίζονται με μορφή ραβδογράμματος κάτω από το διάγραμμα της τιμής, με πράσινο χρώμα σε ανοδικές συνεδριάσεις και με κόκκινο σε πτωτικές (Εικόνα 5-2).

Ταυτόχρονα με τα διαγράμματα τιμής και όγκου, εμφανίζεται λίστα επιλογών με τους διαθέσιμους τεχνικούς δείκτες. Επιλέγοντας οποιονδήποτε δείκτη εμφανίζονται οι διαθέσιμες παράμετροι με τις προεπιλεγμένες τιμές τους και στα δεξιά προστίθεται το διάγραμμα για το συγκεκριμένο δείκτη (Εικόνα 5-3).

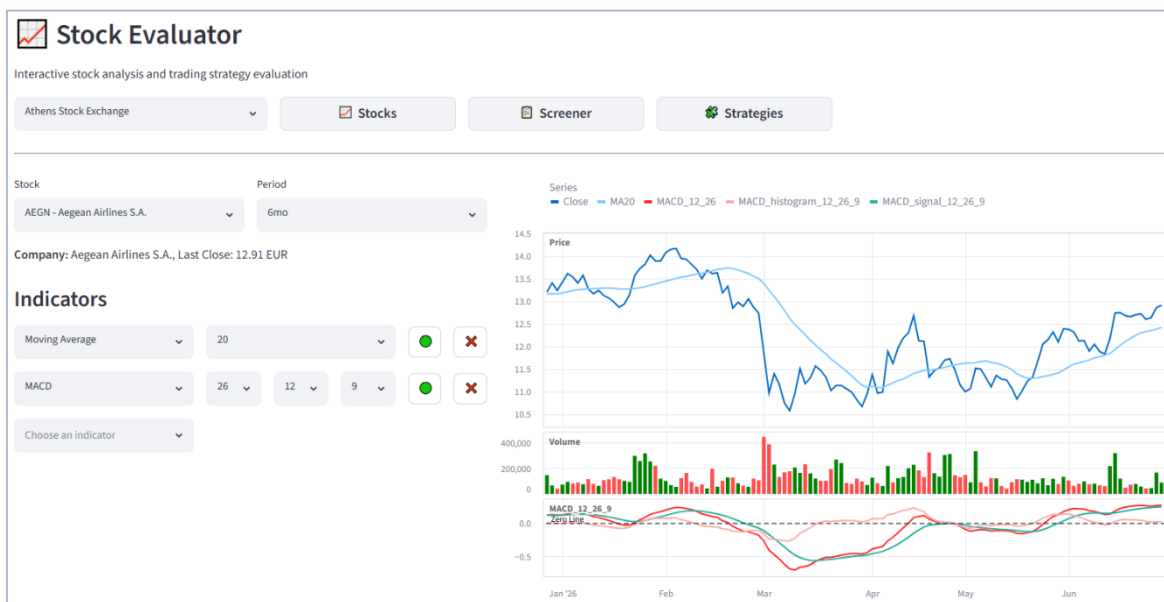


Εικόνα 5-2: Stock Evaluator – Επιλογή μετοχής, και διάγραμμα τιμής / όγκου



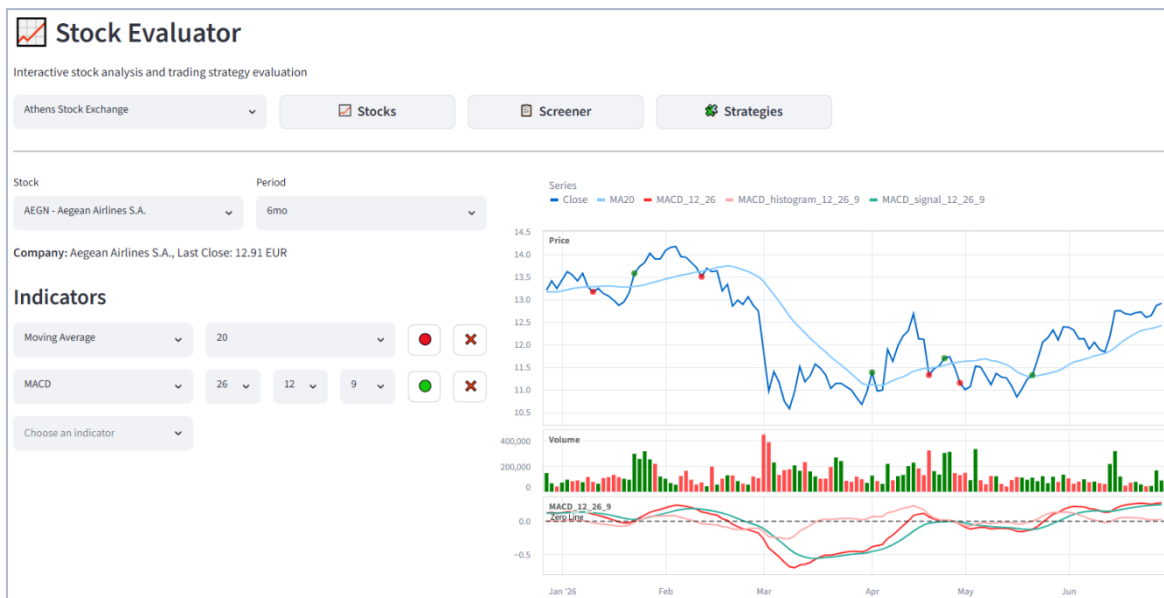
Εικόνα 5-3: Stock Evaluator – Προσθήκη κινητού μέσου όρου 20 ημερών

Ταυτόχρονα εμφανίζεται νέα λίστα επιλογών, και ο χρήστης μπορεί να προσθέσει και άλλους δείκτες με παρόμοιο τρόπο. Ο απλός και ο εκθετικός κινητός μέσος όρος προβάλλονται στο ίδιο διάγραμμα με την τιμή, ενώ οι δείκτες Momentum, MACD και RSI προστίθενται σε νέα, ανεξάρτητα διαγράμματα κάτω από το διάγραμμα όγκου. Στην Εικόνα 5-4 έχει προστεθεί ο δείκτης MACD με παραμέτρους 26, 12, 9.



Εικόνα 5-4: Stock Evaluator – Προσθήκη MACD 26, 12, 9

Δεξιά από κάθε επιλεγμένο δείκτη διατίθεται επιλογή για την εκτέλεση της προκαθορισμένης στρατηγικής για αυτόν τον δείκτη. Στην περίπτωση αυτή τα σήματα αγοράς και πώλησης εμφανίζονται ως μικροί κύκλοι πάνω στο διάγραμμα της τιμής με πράσινο και κόκκινο χρώμα, αντίστοιχα (Εικόνα 5-5).



Εικόνα 5-5: Stock Evaluator – Σήματα αγοράς και πώλησης

Επιλέγοντας Screener στην κορυφή της σελίδας, ο χρήστης μεταφέρεται στη σελίδα αναζήτησης και ταξινόμησης μετοχών με βάση τα θεμελιώδη μεγέθη τους. Πάνω αριστερά εμφανίζονται κριτήρια επιλογής, και στο κάτω μέρος πίνακας αποτελεσμάτων, ο οποίος αρχικά περιλαμβάνει όλες τις μετοχές του επιλεγμένου χρηματιστηρίου (Εικόνα 5-6).

Stock Evaluator

Interactive stock analysis and trading strategy evaluation

Athens Stock Exchange

Stocks

Screener

Strategies

Criteria

Sector

Choose options

Industry

Choose options

Market cap (min)

0

-

+

Market cap (max)

0

-

+

Dividend yield % (min)

0.00

-

+

P/E (max)

0.00

-

+

Min 52w Performance (%)

0.00

-

+

Max 52w Position (%)

0.00

-

+

Selected stocks

No stocks selected.

Strategy back testing

Results

☐ Select / unselect all matching stocks

☐	Symbol	Name	Last Close	Sector	Industry	Market Cap	Dividend Yield %	P/E	EPS	Avg Volume	52w %	52w Low	52w High	52w Pos
☐	AAAK	Wool Industry Tria Alfa S.A.	6.25	Consumer Cyclical	Textile Manufacturing	6,318,750	None	None	-0.15	50	-12.59	4.64	9.00	
☐	ADMIE	Admie Holding S.A.	4.61	Utilities	Utilities - Independent Power	1,135,741,568	3.74	8.45	0.58	994,780	54.57	2.75	4.90	
☐	AEGN	Aegean Airlines S.A.	12.86	Industrials	Airlines	1,158,496,512	6.99	7.82	1.65	123,613	5.82	10.80	15.36	
☐	AIA	ATHENS INTERNATIONAL AIRPORT	10.57	Industrials	Airports & Air Services	3,413,349,120	6.17	15.78	0.68	156,538	8.66	9.68	12.00	
☐	AKTR	Aktor Societe Anonyme Holding Company Technical and Energy I	13.36	Industrials	Engineering & Construction	2,729,151,488	None	148.67	0.09	264,600	148.70	5.31	13.68	

Εικόνα 5-6: Stock Evaluator – Σελίδα Screener

Ο χρήστης μπορεί να επιλέξει μετοχές με βάση πολλαπλούς τομείς (sectors) και βιομηχανίες (industries), εύρος κεφαλαιοποίησης, ελάχιστη μερισματική απόδοση, μέγιστο P/E, ελάχιστη απόδοση τελευταίων 52 εβδομάδων, καθώς και μέγιστη ποσοστιαία θέση της τρέχουσας τιμής στο εύρος τιμών αυτού του διαστήματος (για παράδειγμα μετοχές που βρίσκονται το πολύ στο 80% της μέγιστης τιμής 52 εβδομάδων).

Η λίστα αποτελεσμάτων ανανεώνεται αυτόματα με κάθε αλλαγή κριτηρίων από το χρήστη, και μπορεί να ταξινομηθεί κάνοντας κλικ σε οποιαδήποτε στήλη. Στο παράδειγμα που ακολουθεί εμφανίζονται οι μετοχές πρώτων υλών και ενέργειας, που έχουν ταξινομηθεί με βάση την κεφαλαιοποίησή τους (Εικόνα 5-7)

Stock Evaluator
 Interactive stock analysis and trading strategy evaluation

Athens Stock Exchange

Stocks

Screener

Strategies

Criteria
 Sector: Basic Materials x Energy x
 Industry: Choose options
 Market cap (min): 0 - + Market cap (max): 0 - + Dividend yield % (min): 0.00 - +
 P/E (max): 0.00 - + Min 52w Performance (%): 0.00 - + Max 52w Position (%): 0.00 - +

Selected stocks
 No stocks selected.
 Strategy back testing

☐ Select / unselect all matching stocks

Symbol	Name	Last Close	Sector	Industry	Market Cap	Dividend Yield %	P/E	EPS	Avg Volume	52w %	52w Low	52w High	52w Position %	
☐	MOH	Motor Oil (Hellas) Corinth Refineries S.A.	39.40	Energy	Oil & Gas Refining & Marketing	4,131,861,760	4.68	4.62	8.25	179,841	61.20	23.44	40.70	92.47
☐	ELPE	HELLENIQ ENERGY Holdings S.A.	10.66	Energy	Oil & Gas Refining & Marketing	3,361,987,072	5.46	7.48	1.47	290,806	33.66	7.55	11.36	81.60
☐	PLAKR	Plastika Kritis S.A.	14.40	Basic Materials	Specialty Chemicals	394,260,480	3.82	10.29	1.40	224	2.13	13.20	17.20	30.00
☐	ALMY	Alumil Aluminium Industry S.A.	6.05	Basic Materials	Aluminum	193,509,664	None		0.00	33,056	19.40	4.66	6.88	62.61
☐	DAIOS	Daios Plastics S.A.	9.90	Basic Materials	Specialty Chemicals	147,900,000	None	14.94	0.66	2,567	150.25	3.84	11.30	81.23

Εικόνα 5-7: Stock Evaluator – Αναζήτηση και ταξινόμηση μετοχών

Από τη λίστα των αποτελεσμάτων, ο χρήστης μπορεί να επιλέξει μετοχές κάνοντας κλικ στο checkbox που εμφανίζεται στην πρώτη στήλη, και οι επιλεγμένες μετοχές συγκεντρώνονται στη λίστα που εμφανίζεται πάνω δεξιά. Από εκεί ο χρήστης μπορεί να τις χρησιμοποιήσει ως είσοδο στη λειτουργία Strategy back testing, που παρέχεται μέσω της σελίδας Strategies (Εικόνα 5-8). Με άλλα λόγια, ο χρήστης αφού επιλέξει μετοχές που ικανοποιούν τα κριτήρια αξιολόγησής του, μπορεί να ελέγξει τη συμπεριφορά των μετοχών αυτών απέναντι σε συγκεκριμένες στρατηγικές συναλλαγών.

Stock Evaluator
 Interactive stock analysis and trading strategy evaluation

Athens Stock Exchange

Stocks

Screener

Strategies

Criteria
 Sector: Basic Materials x Energy x
 Industry: Choose options
 Market cap (min): 0 - + Market cap (max): 0 - + Dividend yield % (min): 0.00 - +
 P/E (max): 0.00 - + Min 52w Performance (%): 0.00 - + Max 52w Position (%): 0.00 - +

Selected stocks
 MOH - Motor Oil (Hellas) Corinth Refineries S.A. ☐
 ELPE - HELLENIQ ENERGY Holdings S.A. ☐
 ALMY - Alumil Aluminium Industry S.A. ☐
 Strategy back testing

☐ Select / unselect all matching stocks

Symbol	Name	Last Close	Sector	Industry	Market Cap	Dividend Yield %	P/E	EPS	Avg Volume	52w %	52w Low	52w High	52w Position %	
☒	MOH	Motor Oil (Hellas) Corinth Refineries S.A.	39.40	Energy	Oil & Gas Refining & Marketing	4,131,861,760	4.68	4.62	8.25	179,841	61.20	23.44	40.70	92.47
☒	ELPE	HELLENIQ ENERGY Holdings S.A.	10.66	Energy	Oil & Gas Refining & Marketing	3,361,987,072	5.46	7.48	1.47	290,806	33.66	7.55	11.36	81.60
☐	PLAKR	Plastika Kritis S.A.	14.40	Basic Materials	Specialty Chemicals	394,260,480	3.82	10.29	1.40	224	2.13	13.20	17.20	30.00
☒	ALMY	Alumil Aluminium Industry S.A.	6.05	Basic Materials	Aluminum	193,509,664	None		0.00	33,056	19.40	4.66	6.88	62.61
☐	DAIOS	Daios Plastics S.A.	9.90	Basic Materials	Specialty Chemicals	147,900,000	None	14.94	0.66	2,567	150.25	3.84	11.30	81.23

Εικόνα 5-8: Stock Evaluator – Επιλογή μετοχών για strategy back testing

Πατώντας στο κουμπί Strategy back testing στο Screener, ή επιλέγοντας Strategies από το κεντρικό μενού, ο χρήστης μεταφέρεται στη σελίδα εκτέλεσης και δοκιμής στρατηγικών. Στα αριστερά εμφανίζεται λίστα με τις επιλεγμένες μετοχές (ήδη συμπληρωμένη όταν ο χρήστης κατευθύνεται σε αυτή τη σελίδα από το Screener) με τα κατάλληλα widgets για προσθήκη ή αφαίρεση μετοχών. Στα δεξιά γίνεται επιλογή στρατηγικής και καθορισμός παραμέτρων εκτέλεσης (Εικόνα 5-9).

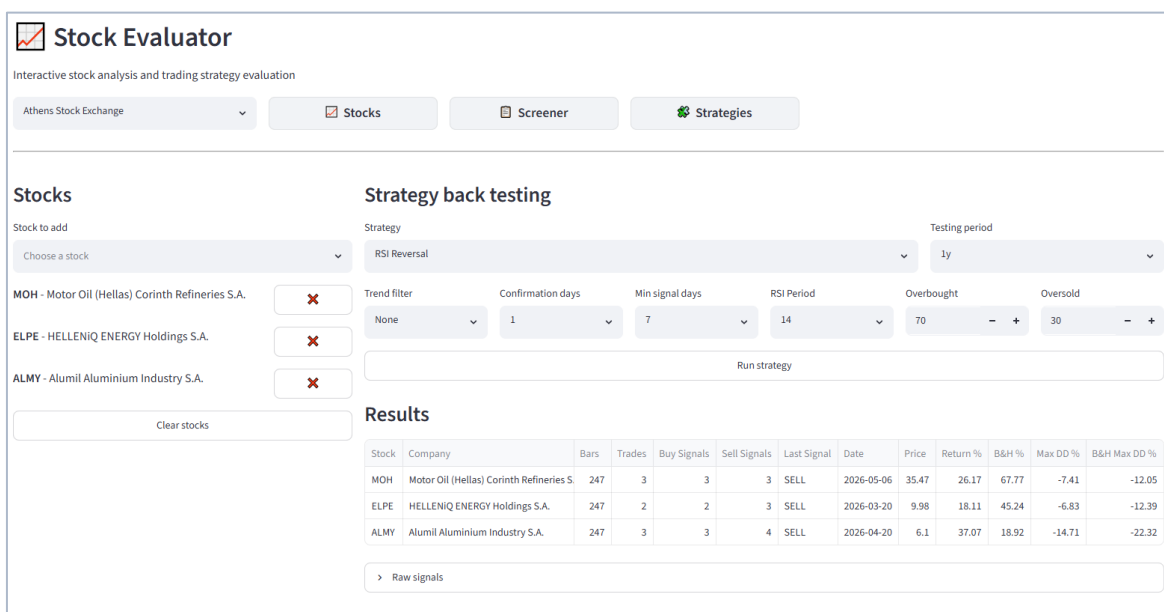
The screenshot shows the 'Stock Evaluator' application interface. At the top, there's a navigation bar with 'Stocks', 'Screener', and 'Strategies' tabs. The 'Strategies' tab is active. Below the navigation bar, there's a section titled 'Strategy back testing'. On the left, under 'Stocks', there's a list of stocks: 'MOH - Motor Oil (Hellas) Corinth Refineries S.A.', 'ELPE - HELLENIC ENERGY Holdings S.A.', and 'ALMY - Alumi Aluminium Industry S.A.'. Each stock has a red 'X' button next to it. Below the list is a 'Clear stocks' button. On the right, under 'Strategy back testing', there's a 'Strategy' dropdown menu, a 'Testing period' dropdown menu set to '1y', and a 'Run strategy' button. A blue banner at the bottom of the strategy section says 'Select stocks and a strategy, then run the strategy.'

Εικόνα 5-9: Stock Evaluator – Σελίδα Strategies

Επειδή η εκτέλεση στρατηγικών είναι χρονοβόρα διαδικασία, τα αποτελέσματα δεν ανανεώνονται αυτόματα με την αλλαγή παραμέτρων (σε αντίθεση με τις σελίδες Stocks και Screener), αλλά ο χρήστης πρέπει να εκτελέσει την επιλεγμένη στρατηγική πατώντας το κουμπί Run Strategy. Τα αποτελέσματα εμφανίζονται σε πίνακα στο κάτω μέρος της σελίδας.

Στο παράδειγμα που ακολουθεί φαίνονται τα αποτελέσματα της στρατηγικής RSI Reversal για διάστημα ενός έτους, περίοδο 14 ημερών, επίπεδα overbought/oversold 70/30, και ελάχιστη απόσταση μεταξύ σημάτων 7 ημέρες (Εικόνα 5-10).

Όπως σε όλους τους πίνακες που εμφανίζονται στην εφαρμογή, ο χρήστης μπορεί να ταξινομήσει τα αποτελέσματα κάνοντας κλικ σε οποιαδήποτε στήλη, για παράδειγμα να ταξινομήσει τις μετοχές με βάση την απόδοση, τη μέγιστη πτώση στην χρονική περίοδο της δοκιμής, ή τον αριθμό των συναλλαγών. Μπορεί επίσης να εξάγει τα δεδομένα σε αρχείο .CSV.



Εικόνα 5-10: Stock Evaluator – Back testing στρατηγικής

5.2. Αξιολόγηση της εφαρμογής και πιθανές βελτιώσεις

Αξιολογώντας την εφαρμογή ως μια πειραματική υλοποίηση στα πλαίσια της εργασίας, τα αποτελέσματα κρίνονται συνολικά ικανοποιητικά, καθώς επιβεβαιώνουν τις βασικές αρχιτεκτονικές και σχεδιαστικές επιλογές, καλύπτουν σε μεγάλο βαθμό τον κύριο στόχο της επεκτασιμότητας, και αναδεικνύουν μια έγκυρη προσέγγιση στην ανάπτυξη αυτού του είδους εφαρμογών στο οικοσύστημα της Python.

Πιο συγκεκριμένα, από την πλευρά της αρχιτεκτονικής και της γενικότερης προσέγγισης στην υλοποίηση της εφαρμογής, οι παρατηρήσεις που έγιναν για το Streamlit στο Κεφάλαιο 3 επιβεβαιώθηκαν στην πράξη, καθώς η εφαρμογή αναπτύχθηκε δαπανώντας ελάχιστο χρόνο στο frontend, και καθόλου χρόνο για αντιμετώπιση ευρύτερων ζητημάτων αρχιτεκτονικής, διαχείρισης HTTP αιτημάτων, ή δρομολόγησης. Το Streamlit δεν επέβαλε ουσιαστικούς περιορισμούς στο υπολογιστικό κομμάτι της εφαρμογής και το ζήτημα της απόδοσης ή της ταχύτητας απόκρισης αντιμετωπίστηκε επιτυχώς με το μηχανισμό caching και τη χρήση του session state. Από την άλλη πλευρά, διαπιστώθηκε ότι οι γνωστοί περιορισμοί του Streamlit στον ακριβή έλεγχο του UI μεγεθύνονται σε κάποιο βαθμό όταν το layout προκύπτει δυναμικά, όπως στην περίπτωση της παραμετροποίησης των δεικτών και των στρατηγικών. Στα σημεία αυτά, καθώς και στο layout του κεντρικού μενού, χρειάστηκαν ορισμένες επεμβάσεις ή παρεκκλίσεις από την τυπική χρήση του Streamlit, για τη σωστή στοίχιση, την καλύτερη εκμετάλλευση του χώρου, τη συμπεριφορά των επιλογών, και τη συνολική εργονομία της εφαρμογής.

Με βάση τις απλοποιήσεις και τις παραδοχές που έχουν γίνει σε αυτήν την έκδοση, ο κεντρικός στόχος της επεκτασιμότητας επιτεύχθηκε πλήρως στους τεχνικούς δείκτες και τις στρατηγικές συναλλαγών, καθώς οι νέες υλοποιήσεις απαιτούν ελάχιστο κώδικα με μειωμένο κίνδυνο εισαγωγής σφαλμάτων στην υπάρχουσα λειτουργικότητα (regression risk). Το βασικό μοντέλο κλάσεων που περιγράφηκε στις παραγράφους 4.4 – 4.6 κρίνεται ότι είναι στη σωστή κατεύθυνση, και μπορεί στο μέλλον να διευρυνθεί περαιτέρω για να καλύψει πιο σύνθετες περιπτώσεις ή πιο προχωρημένες υλοποιήσεις, χωρίς να ανασχεδιαστεί ριζικά. Η χρήση αντικειμενοστραφούς προσέγγισης σε αυτό το κομμάτι της υλοποίησης θεωρείται ότι βοήθησε σε αυτήν την κατεύθυνση. Η αναζήτηση εταιρειών και το back testing των στρατηγικών έχουν σχεδιαστεί με πιο απλό τρόπο ο οποίος, ενώ καλύπτει επαρκώς τους στόχους της εργασίας, θα μπορούσε να αναθεωρηθεί σε μια μελλοντική έκδοση.

Ως προς την λειτουργικότητα της εφαρμογής, οι στόχοι έχουν επιτευχθεί σε ικανοποιητικό βαθμό, με κάποιες προφανείς βελτιώσεις στον τομέα της αξιολόγησης των θεμελιωδών μεγεθών, που θα μπορούσαν να υλοποιηθούν μελλοντικά. Στην περίπτωση αυτή, τα θεμελιώδη μεγέθη που χρησιμοποιούνται για την αρχική αναζήτηση και ταξινόμηση των μετοχών θα εντάσσονταν σε ένα πιο συστηματικό μοντέλο, με σαφέστερη διάκριση μεταξύ κριτηρίων αναζήτησης και κανόνων αξιολόγησης.

Σε σχέση με την ορθότητα των υπολογισμών, οι δείκτες έχουν ελεγχθεί συγκρίνοντας τα αποτελέσματα με άλλα εργαλεία και online πλατφόρμες, και οι στρατηγικές έχουν επίσης ελεγχθεί δειγματοληπτικά. Σε περίπτωση περαιτέρω ανάπτυξης της εφαρμογής θα ήταν σκόπιμο να υλοποιηθούν αυτοματοποιημένοι έλεγχοι.

Η ευρύτερη λειτουργικότητα των στρατηγικών είναι πιο σύνθετη στην αξιολόγησή της. Σε πολλές περιπτώσεις, οι στρατηγικές που υλοποιήθηκαν εμφανίζουν χαμηλότερη απόδοση από μια απλή τακτική αγοράς και διακράτησης (buy & hold). Το αποτέλεσμα αυτό είναι σε κάποιο βαθμό αναμενόμενο, ιδίως σε μετοχές με ισχυρή μακροπρόθεσμη τάση, και εξισορροπείται μερικώς από το γεγονός ότι οι στρατηγικές με χαμηλότερη απόδοση έχουν συχνά και χαμηλότερη “μέγιστη πτώση” (max drawdown), πράγμα που μειώνει το ρίσκο που αναλαμβάνει ο επενδυτής διατηρώντας μια θέση. Ενδεχομένως όμως το ότι οι στρατηγικές της εφαρμογής συχνά υποαποδίδουν είναι και μια ένδειξη ότι η υλοποίηση πιθανόν απαιτεί πιο σύνθετη προσέγγιση, ιδιαίτερα στα φίλτρα επιβεβαίωσης και τις συνθήκες ανοίγματος ή κλεισίματος μιας θέσης. Σημειώνεται επίσης ότι στην εφαρμογή δεν λαμβάνονται υπόψη τα κόστη συναλλαγών, τα οποία επιβαρύνουν περαιτέρω την απόδοση σε συχνές αγορές και πωλήσεις. Συμπερασματικά, από όλες τις πιθανές βελτιώσεις που αναφέρθηκαν ο εμπλουτισμός των στρατηγικών με πρόσθετα χαρακτηριστικά και η

υλοποίηση πιο σύνθετων αλγορίθμων εισόδου και εξόδου θα ήταν από τις πρώτες προτεραιότητες σε ενδεχόμενη μελλοντική εργασία.

Ιδιαίτερο ενδιαφέρον θα είχε επίσης η ενσωμάτωση μεθόδων Τεχνητής Νοημοσύνης, για παράδειγμα στην αναγνώριση διαγραμματικών σχηματισμών (patterns), όπως αυτών που περιγράφηκαν στην παράγραφο 2.3.

Ανακεφαλαιώνοντας, η εφαρμογή θεωρείται ότι πέτυχε τους στόχους της, ιδιαίτερα σε ό,τι αφορά το σχεδιασμό και την υλοποίηση της βασικής λειτουργικότητας καθώς και την επεκτασιμότητα. Με σημείο εκκίνησης τον κώδικα που έχει αναπτυχθεί, μπορεί κανείς να πειραματιστεί με την αξιολόγηση μετοχών στο οικοσύστημα της Python, χρησιμοποιώντας στοιχεία θεμελιώδους και τεχνικής ανάλυσης, με διαδεδομένα εργαλεία που είναι δημόσια διαθέσιμα και καλά τεκμηριωμένα, καθώς και να επεκτείνει ή να βελτιώσει την εφαρμογή στους τομείς που αναφέρθηκαν.

Βιβλιογραφία

Ακολουθούν οι βιβλιογραφικές αναφορές (πηγές) της Εργασίας.

Brock, W., Lakonishok, J. and LeBaron, B. (1992). Simple Technical Trading Rules and the Stochastic Properties of Stock Returns, *The Journal of Finance*, 47(5), pp. 1731–1764.
<https://doi.org/10.1111/j.1540-6261.1992.tb04681.x>

Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Boston: Addison-Wesley.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Garrett, J.J. (2005). Ajax: A New Approach to Web Applications, *Adaptive Path*, 18/02/2005. Ανακτήθηκε από: https://designftw.mit.edu/lectures/apis/ajax_adaptive_path.pdf (ημερομηνία επίσκεψης: 12/06/2026)

McKinney, W. (2010). Data Structures for Statistical Computing in Python, *Proceedings of the 9th Python in Science Conference*, pp. 56–61. <https://doi.org/10.25080/Majora-92bf1922-00a>.

Murphy, J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. New York Institute of Finance. Διαθέσιμο στο: <https://archive.org/details/JohnJ.MurphyTechnicalAnalysisOfTheFinancialMarkets>

pandas Documentation v.3.0.3. (2026). Ανακτήθηκε από: <https://pandas.pydata.org/docs/> (ημερομηνία επίσκεψης: 14/05/2026)

Park, C.-H. and Irwin, S.H. (2007). What Do We Know About the Profitability of Technical Analysis, *Journal of Economic Surveys*, 21(4), pp. 786–826.
<https://doi.org/10.1111/j.1467-6419.2007.00519.x>

SIFMA Foundation (2020). *Market Structure Download*, SIFMA Insights., 08/10/2020. Ανακτήθηκε από <https://www.sifma.org/research/insights/market-structure-download>

Streamlit documentation v.1.58.0 (2026). Ανακτήθηκε από: <https://docs.streamlit.io/> (ημερομηνία επίσκεψης: 02/04/2026)

TIOBE (2026). *The TIOBE Programming Community index*. Ανακτήθηκε από: <https://www.tiobe.com/tiobe-index/> (ημερομηνία επίσκεψης: 17/06/2026)

Vega-Altair documentation v.6.2.1. (2026). Ανακτήθηκε από: <https://altair-viz.github.io/> (ημερομηνία επίσκεψης: 23/06/2026)

Yahoo Finance (2026). Ανακτήθηκε από: <https://finance.yahoo.com/> (ημερομηνία επίσκεψης: 18/03/2026)

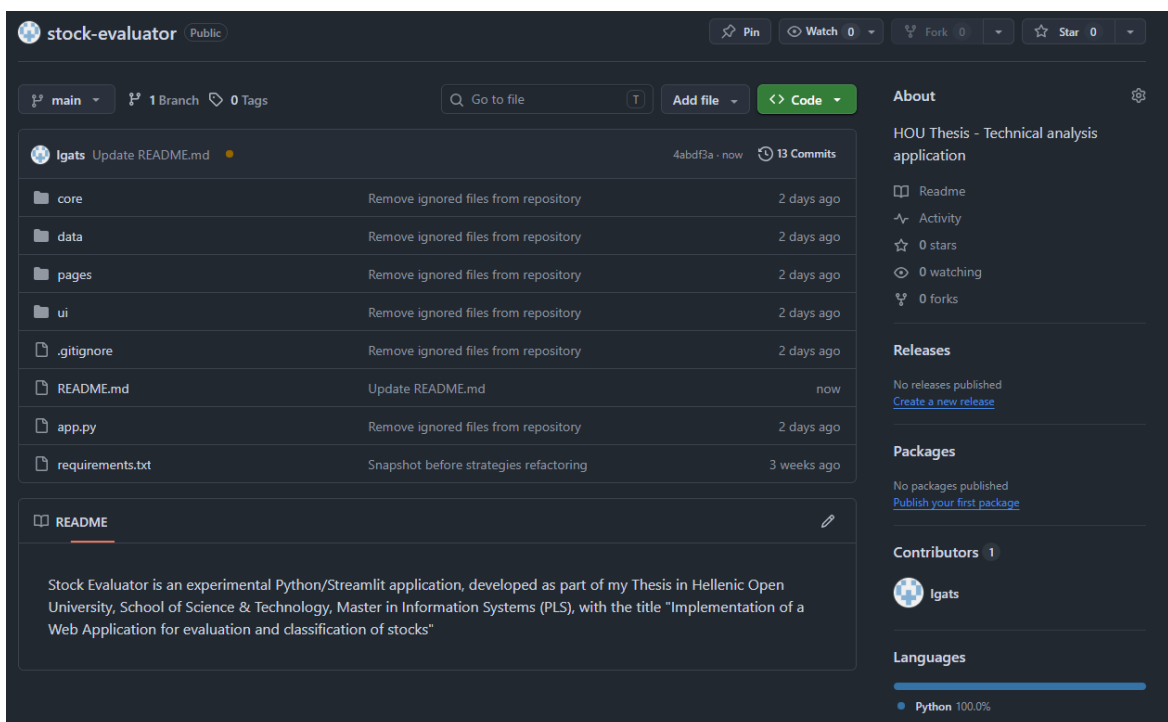
yfinance GitHub repository (2026). Ανακτήθηκε από: <https://github.com/ranaroussi/yfinance> (ημερομηνία επίσκεψης: 04/05/2026)

Παράρτημα Α: GitHub Repository

Ο κώδικας της εφαρμογής φιλοξενείται στο GitHub, σε δημόσιο αποθετήριο που επιτρέπει ευκολότερη διάθεση για μελέτη ή επαναχρησιμοποίηση από τρίτους.

Το αποθετήριο βρίσκεται στη διεύθυνση <https://github.com/Igats/stock-evaluator> και περιλαμβάνει το αρχείο README.md, το οποίο περιγράφει συνοπτικά το σκοπό και τις λειτουργίες της εφαρμογής.

Τα περιεχόμενα του αποθετηρίου είναι οργανωμένα στους τέσσερις φακέλους που περιγράφηκαν στην παράγραφο 4.1, όπως φαίνεται στην εικόνα.



Παράρτημα Β: Χρήση Τεχνητής Νοημοσύνης

Κατά την συγγραφή της Διπλωματικής Εργασίας, χρησιμοποιήθηκαν εργαλεία Τεχνητής Νοημοσύνης επικουρικά, για πειραματισμό και δημιουργία προσχεδίων για τα UML class diagrams που περιλαμβάνονται στις παραγράφους 4.4 – 4.6 και το UML sequence diagram που δίνεται στο Σχήμα 3-2.

Για τα class diagrams, ο κώδικας από το πακέτο core τροφοδοτήθηκε αρχικά στο ChatGPT (GPT-5.5) και ζητήθηκε η παραγωγή των διαγραμμάτων. Έγιναν δοκιμές για απευθείας παραγωγή εικόνων, παραγωγή διαγραμμάτων τύπου draw.io, τύπου Mermaid, και PlantUML, και τελικά αποφασίστηκε η χρήση της μορφής PlantUML.

Επειδή η αναλυτική απεικόνιση όλων των κλάσεων, των properties και των attributes της Python δεν εξυπηρετεί τους σκοπούς της εργασίας, έγιναν αρκετοί πειραματισμοί στο ChatGPT, παράγοντας τα διαγράμματα με διαφορετικές κάθε φορά συμβάσεις, μέχρι να έρθουν στην τελική μορφή που παρουσιάζεται στο Κεφάλαιο 4.

Με παρόμοιο τρόπο, αλλά με οδηγίες αντί για χρήση κώδικα, δημιουργήθηκε το sequence diagram που δίνεται στο Σχήμα 3-2.

Το rendering των τελικών PlantUML αρχείων έγινε με τον “Online Server” που διατίθεται στο www.plantuml.com.

Υπεύθυνη Δήλωση Συγγραφέα:

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν.1599/1986, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης.