Analysis and Requirements Report

T2402

Presented by:
Emre Akgül, Uygar Aras, Asım Adil Can, Gün Taştan, Alara Zeybek

15.12.2024

1. Introduction

Plutos Equities is a financial and capital forecasting platform designed to predict the quarterly financial performance of the top 100 NASDAQ-listed companies. The primary objective of this project is to support financial stakeholders, including investors, auditors, and analysts, with reliable predictions of key financial metrics such as revenue, net income, and operating expenses. By delivering forecasts before quarterly reports are officially released, Plutos Equities provides users with a tool to make data-driven decisions, assess risks, and identify opportunities.

2. Current System

Currently, there are no automated solutions specifically designed for capital forecasting. Existing systems mostly focus on stock price prediction, with platforms like Bloomberg Terminal, AlphaSense, and TradingView using historical data, technical indicators, and sentiment analysis to forecast market trends and price movements. However, these tools do not address the prediction of capital metrics such as revenue, net income, and operating expenses. On the other hand, capital forecasting remains a largely manual process conducted by financial analysts and auditors. These professionals rely on tools like Excel to analyze financial statements, economic indicators, and market trends, which, while effective for detailed analysis, are time-intensive, and lack the scalability that an automated solution could provide. As a result, there is no direct alternative to an automated capital forecasting system like Plutos Equities.

3. Proposed System

3.1. Overview

Plutos Equities focuses on providing a comprehensive financial forecasting solution by predicting the quarterly reports of the top 100 NASDAQ-listed companies [1]. The project aims to offer accurate, data-driven forecasts of critical financial metrics for the next quarter. These forecasts include revenue, expenses, operating income, and other key metrics that are important for understanding a company's financial position.

The platform combines structured data, such as historical financial statements and market indices, with unstructured data from news articles, earnings call transcripts, and social media sentiment. Machine learning models are used to process these data sources, uncovering patterns and trends that traditional methods often miss [2]. The result is a set of predictions that are not only accurate but also insightful for stakeholders looking to make informed decisions.

To give an example, the platform can forecast Apple Inc.'s Q1 2025 performance by analyzing its financial data from previous quarters, combined with real-time updates during the three months leading up to the new quarter. These updates might include news of a new product launch, changes in production costs, or shifts in global economic factors that impact the tech industry. By using both historical and recent data, the platform provides a clear prediction of Apple's revenue, expenses, and profitability, helping stakeholders anticipate the company's performance ahead of official earnings reports and quarter filings [3].

Unlike traditional stock price forecasting tools, which often fail to provide a nuanced understanding of a company's underlying financial health, Plutos Equities focuses on the core metrics that define corporate performance. The platform achieves this by combining data collection methods, feature engineering, and machine learning models. The integration of

real-time data sources, such as financial news and earnings call transcripts, further enhances the predictions and allows for a dynamic and adaptive forecasting system.

In addition to its forecasting capabilities, Plutos Equities features a user-friendly interface that allows users to explore predictions through customizable dashboards and visualizations. This ensures that both institutional investors and individual users can easily access and interpret the data, regardless of their level of financial expertise.

By focusing on financial metrics rather than stock prices, Plutos Equities provides a more detailed view of company performance and health, making it a valuable resource for anyone involved in the financial analysis of NASDAQ companies.

Our project is going to be within the scope of the product system, which is under the experience section. The innovation we are bringing to the market is focused on transparency and accessibility in financial forecasting. Unlike existing tools that offer broad financial data aggregation or limited stock price predictions, Plutos Equities is uniquely positioned as the first platform to provide transparent quarterly financial forecasting for NASDAQ companies, openly sharing its data sources. By detailing the sources of our predictions—such as financial statements, market indices, real-time news, and social sentiment—our project builds trust and credibility, offering investors, auditors, and other stakeholders a reliable and explainable solution.

This approach addresses a gap in the market, as most existing tools operate as "black boxes," providing users with a limited understanding of how their forecasts are generated. By delivering a platform that clearly communicates its data sources, Plutos Equities provides a straightforward and reliable financial forecasting solution [4]. This transparency allows users to better understand the predictions and use them effectively in their decision-making processes. The platform aims to change how financial forecasting is approached, making it practical and accessible to a broad range of users, including investors, auditors, and corporate decision-makers.

Despite its innovative approach, a key challenge lies in integrating and updating diverse data sources in real-time [5]. As financial data is inherently dynamic, incorporating unstructured inputs like earnings call transcripts or social sentiment into our predictive models without compromising accuracy is a complex task.

3.2. Functional Requirements

3.2.1. Quarterly Financial Prediction Functionality

- Generate predictions for key financial metrics (Revenue, EPS, Gross Margin) for NASDAQ Top 100 companies.
- Review confidence intervals for each prediction.
- Forecast financial performance for the upcoming quarter.
- Potentially extend predictions up to 3 quarters in the future.

3.2.2. User Functionalities

- Support user account creation/authentication
    - Users can create an account.
        - Login/logout functionalities.

        - Change membership plan.
        - Change the payment preferences.
    - Follow companies you are interested in.
    - Unfollow companies.
    - Allow personalization of the dashboard and metrics.
        - Customize/edit financial charts according to ease of use.
    - Provide insights on prediction interpretation.
    - Implement feedback mechanisms for continuous improvement.
    - Visualization of simple parameters about companies and their stocks.
    - Visualization of the predicted quarter report about companies.
    - Allow users to export overall forecast results in different formats (.csv, .xls, etc.)

3.2.3. Admin Functionalities

- Provide a centralized admin dashboard for system monitoring, configuration, and management.
- Comprehensive documentation, including system architecture, user manuals, and troubleshooting guides.

3.3. Nonfunctional Requirements

3.3.1. Performance

The platform will process a wide range of financial data, including historical financial statements, market volatility indicators, economic trends, industry-specific data, regulatory factors, and news sentiment. Therefore, the system must implement efficient database indexing and query optimization to maintain response times. Maximum system load capacity should handle at least 200 simultaneous user sessions and reduce load times for frequently accessed data.

Furthermore, to comply with the business performance standard, the platform will achieve a Mean Absolute Percentage Error (MAPE) of less than 15% for forecasts in its first-to-next financial report. It will also ensure accurate predictions based on its own threshold (examined under Success Metrics in the report) for at least 80% of the NASDAQ Top 100 companies.

3.3.2. Reliability

Plutos Equities uses consistent analytics algorithms across 100 companies and relies on similar methods to obtain up-to-date data, enabling the error handling system to be automated and effectively manage and recover from unforeseen failures without the need for manual intervention. Additionally, the system will maintain at least %99 uptime to ensure availability during critical financial periods, such as quarterly report releases. Servers also save processed data, predictions, and user configurations in a secure AWS-hosted database.

3.3.3. Usability

One of the highlights of Plutos Equities' current financial applications is that it offers an open and accessible service that anyone can use, regardless of their level of knowledge. Accordingly, the system will have an easy-to-use interface that requires new users to receive training for no more than 20 minutes. It will support responsive design to ensure optimal viewing and interaction experience across desktop, tablet, and mobile devices. Interactive charts and graphs (using D3.js or Chart.js) will help users visualize trends, predictions, and anomalies in real time. It will have customizable views based on user preferences and sector focus. Any page the user requests will be available within 2 seconds.

3.3.4. Scalability

Plutos Equities designed the system architecture to scale horizontally by adding more servers or cloud instances without significant reconfiguration. The system will be tested for scenarios that increase traffic by up to 10 times. With AWS, it supports automatic load balancing to distribute user requests efficiently across available resources. Its design is cloud-native, and it has a containerized architecture to facilitate easy scaling and deployment.

3.3.5. Maintainability

Plutos's system is built with maintainability in mind, making it simple to upgrade, change, or expand with little risk or work. To make debugging and future development easier, code follows established naming standards, modular design principles, and thorough documentation methods. To reduce technological debt, dependencies, and configurations must be properly described and maintained. The system's integrated logging, monitoring, and error-handling features will enable effective problem diagnosis and resolution. To minimize downtime and manual intervention, maintenance procedures, including upgrades, scaling, and deployment, must be automated whenever feasible. In order to guarantee smooth updates, the system must also be backward-compatible with earlier iterations and contain test suites to ensure that modifications do not interfere with already-existing functionality.

3.3.6. Security

Plutos Equities uses AWS servers to enforce secure authentication using AWS Identity and Access Management (IAM). Servers adhere to data privacy regulations (e.g., GDPR, CCPA) with encrypted data storage and secure API endpoints.

3.3.7. Sustainability

Plutos Equities recognizes the importance of environmental sustainability and aims to reduce its ecological footprint by at least %10 compared to the average website. It utilizes cloud providers with strong commitments to renewable energy (AWS). Optimize our algorithms for energy efficiency. Additionally, it delivers all reports and analyses digitally. Encourage users to adopt paperless practices.

3.4. Pseudo Requirements

1. **Programming Language:**
    - The platform will be developed using Python as the primary programming language for backend development and machine learning tasks.

2. **Frameworks and Libraries:**
    - FastAPI for backend RESTful APIs.
    - ReactJS for frontend development.
    - Pandas and NumPy for data manipulation and analysis.
    - Scikit-Learn and TensorFlow/PyTorch for machine learning models.
    - RoBERTa and BERT for sentiment analysis.

3. **Data Sources:**
    - Real-time stock prices: Yahoo Finance API, Alpha Vantage API.
    - Financial Statements: SEC EDGAR, Yahoo Finance.
    - News and Sentiment Data: Bloomberg, Wall Street Journal, Reddit APIs.
    - Economic Indicators: World Bank, FRED, Bureau of Labor Statistics (BLS).

4. **Cloud Infrastructure:**
    - Hosted on AWS, utilizing EC2 for hosting, S3 for data storage, and RDS for database management.

5. **Database:**
    - MySQL for storing user preferences, prediction outputs, and financial data.
    - AWS S3 for raw and backup data storage.

6. **Security Standards:**
    - OAuth 2.0 for user authentication and authorization.
    - Compliance with GDPR for data privacy.

7. **Version Control System:**
    - Git with project management on GitHub.

8. **Development Environment:**
    - Pandas, NumPy, VS Code, and PyCharm for coding, debugging, and analysis.

9. **Model Training and Optimization:**
    - ARIMA, Prophet, LSTM, and ensemble models like Random Forest and XGBoost will be implemented for accurate financial forecasting.
    - Regular backtesting will be performed to ensure performance robustness.

10. **User Interface Requirements:**
    - The platform will be mobile and desktop compatible with a customizable dashboard for personalized views and insights.

11. **Alert System:**
    - Real-time alerts for prediction updates, stock price changes using WebSockets.

12. **Subscription and Payment Integration:**
    - Stripe API for managing user subscriptions and payment processing securely.

3.5. System Models

3.5.1. Scenarios

1. Register/Login

Actor: Investor/Financial Analyst

Entry Condition: The user wants to access the Plutos Equities platform.
Exit Condition: User successfully logs in or registers an account.

Flow of Events:

1. User navigates to the login/register page.
2. User enters credentials (email, password).
3. System verifies credentials.
    - If valid, the user is logged in.
    - If not valid, an error is displayed.

4. If registering, the user provides details (e.g., name, email, password, subscription type).
5. System creates an account and sends a confirmation email.

Alternative Flows:

- **Invalid credentials** → System displays an "Invalid credentials" error message.
- **Forgotten password** → System provides a "reset password" option via email verification.
- **Duplicate email during registration** → System displays "Email already registered."

2. Monitor Previous Predictions

Actor: Investor/Financial Analyst

Entry Condition: User wants to compare historical predictions made by the system.
Exit Condition: System displays the previous predictions.

Flow of Events:

1. User selects the "Monitor Previous Predictions" feature.
2. System retrieves and displays historical predictions for the selected companies.
3. Predictions are shown as tables or graphs.

Alternative Flows:

- **Data unavailable** → System displays "Previous prediction data not found."
- **Network error** → System prompts user to retry after checking connectivity.

3. View Current Financial Predictions

Actor: Investor/Financial Analyst

Entry Condition: User selects the option to view predictions for a company.
Exit Condition: System displays the most recent financial predictions.

Flow of Events:

1. User navigates to the "Current Financial Predictions" dashboard.
2. User selects a company.
3. System retrieves the most recent predictions.
4. System displays predictions in tabular and graphical formats.

Alternative Flows:

- **No predictions available** → System displays "Predictions for this company are not available."

4. Follow/Unfollow Companies

Actor: Investor/Financial Analyst

Entry Condition: User wants to follow or unfollow specific companies.
Exit Condition: The favorite company list is updated.

Flow of Events:

1. User navigates to the company list.
2. User selects a company and clicks "Follow" or "Unfollow."
3. System updates the user's favorite company list.

Alternative Flows:

- **Database update failure** → System displays "Unable to update your favorites. Please try again."

5. View Daily Stock Price

Actor: Investor/Financial Analyst

Entry Condition: User wants to view the current daily stock price for a company.
Exit Condition: System displays the daily stock price.

Flow of Events:

1. User selects a company to view its stock price.
2. System fetches stock price data from the Yahoo Finance API.
3. System displays daily stock price details (e.g., open, close, high, low).

Alternative Flows:

- **API failure** → System displays "Unable to fetch stock prices. Try again later."
- **Data unavailable** → System shows a message: "Stock data for this company is not available."

6. View Previous Predictions Against Outcomes

Actor: Investor/Financial Analyst

Entry Condition: User wants to compare previous predictions with actual reported outcomes.
Exit Condition: System displays comparison results.

Flow of Events:

1. User navigates to the "Compare Predictions vs. Outcomes" feature.
2. User selects a company and timeframe.
3. System retrieves previous predictions and actual reported outcomes.
4. System compares both and displays accuracy metrics and discrepancies.

Alternative Flows:

- **Missing outcome data** → System displays "Outcome data for this timeframe is unavailable."

7. View Historical Data Trends

Actor: Investor/Financial Analyst

Entry Condition: User wants to view historical trends of financial data.
Exit Condition: System displays historical trends graphically.

Flow of Events:

1. User selects "View Historical Data Trends."
2. User selects a company and financial metrics.
3. System retrieves historical data for previous quarters.
4. System displays trends as graphs (quarter-by-quarter analysis).

Alternative Flows:

- **Data unavailable** → System shows "Historical data not found for this company."

8. Add/Remove Companies in Favorite List

Actor: Investor/Financial Analyst

Entry Condition: User wants to update their favorite company list.
Exit Condition: The list is successfully updated.

Flow of Events:

1. User selects a company.
2. User clicks “Add to Favorites” or “Remove from Favorites.”
3. System updates the favorite list.

Alternative Flows:

- **System error** → "Unable to update the favorite list. Try again later."

9. View Overall Health Dashboard of a Company

Actor: Investor/Financial Analyst

Entry Condition: User wants to assess the overall health of a company.
Exit Condition: System displays the health dashboard.

Flow of Events:

1. User selects the "Health Dashboard" feature.
2. User selects a company.
3. System retrieves metrics from the API (e.g., revenue growth, expenses).
4. System displays the health summary in visual formats.

Alternative Flows:

- **Metrics unavailable** → System displays "Health metrics currently unavailable."

10. Set Alerts

Actor: Investor/Financial Analyst

Entry Condition: User wants to set alerts for specific events.
Exit Condition: Alerts are saved.

Flow of Events:

1. User selects "Set Alerts."
2. User configures alert parameters (e.g., new prediction update, mid quarter filings, stock price limits).
3. System saves the settings.
4. System sends notifications when conditions are met.

Alternative Flows:

- **System failure** → "Unable to save alerts. Retry later."

11. Export Data

Actor: Investor/Financial Analyst

Entry Condition: User wants to export data.
Exit Condition: Data is downloaded in the selected format.

Flow of Events:

1. User selects "Export Data."
2. User chooses data type and format (CSV, Excel).
3. System generates and provides the file for download.

Alternative Flows:

- **Export failure** → System displays "Unable to export data. Please try again."

12. Manage Payment and Subscription Tier

Actor: Investor/Financial Analyst

Entry Condition: User wants to update payment or subscription settings.
Exit Condition: Payment is processed, and subscription details are updated.

Flow of Events:

1. User navigates to "Manage Subscription."
2. User selects a subscription plan.

3. System interacts with Stripe API to process payment.
4. Subscription is updated.

Alternative Flows:

- **Payment failure** → "Payment unsuccessful. Check the details and try again."

13. Submit Feedback/Issue

Actor: Investor/Financial Analyst

Entry Condition: User encounters an issue or wants to provide feedback.
Exit Condition: Feedback is submitted.

Flow of Events:

1. User selects "Submit Feedback."
2. User enters details and submits the form.
3. System records the feedback and notifies the administrator.

Alternative Flows:

- **System error** → "Unable to submit feedback. Retry later."

14. Give Feedback/Resolve Opened Issues

Actor: System Administrator

Entry Condition: Administrator accesses the list of submitted issues.
Exit Condition: Issues are resolved.

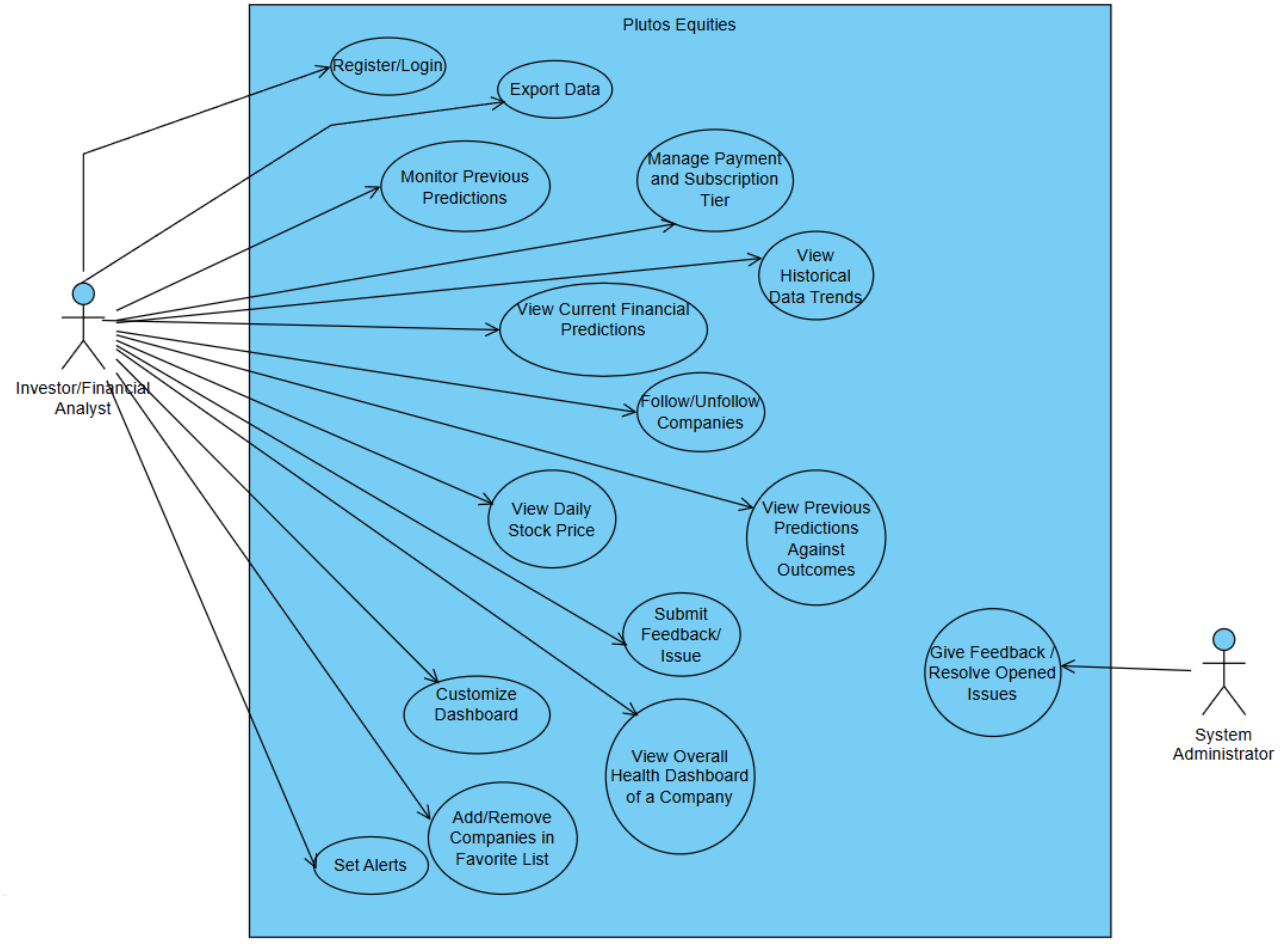
Flow of Events:

1. Administrator logs into the system.
2. Administrator views pending issues.
3. Administrator resolves the issue or provides updates.
4. System marks the issue as "Resolved" and notifies the user.

Alternative Flows:

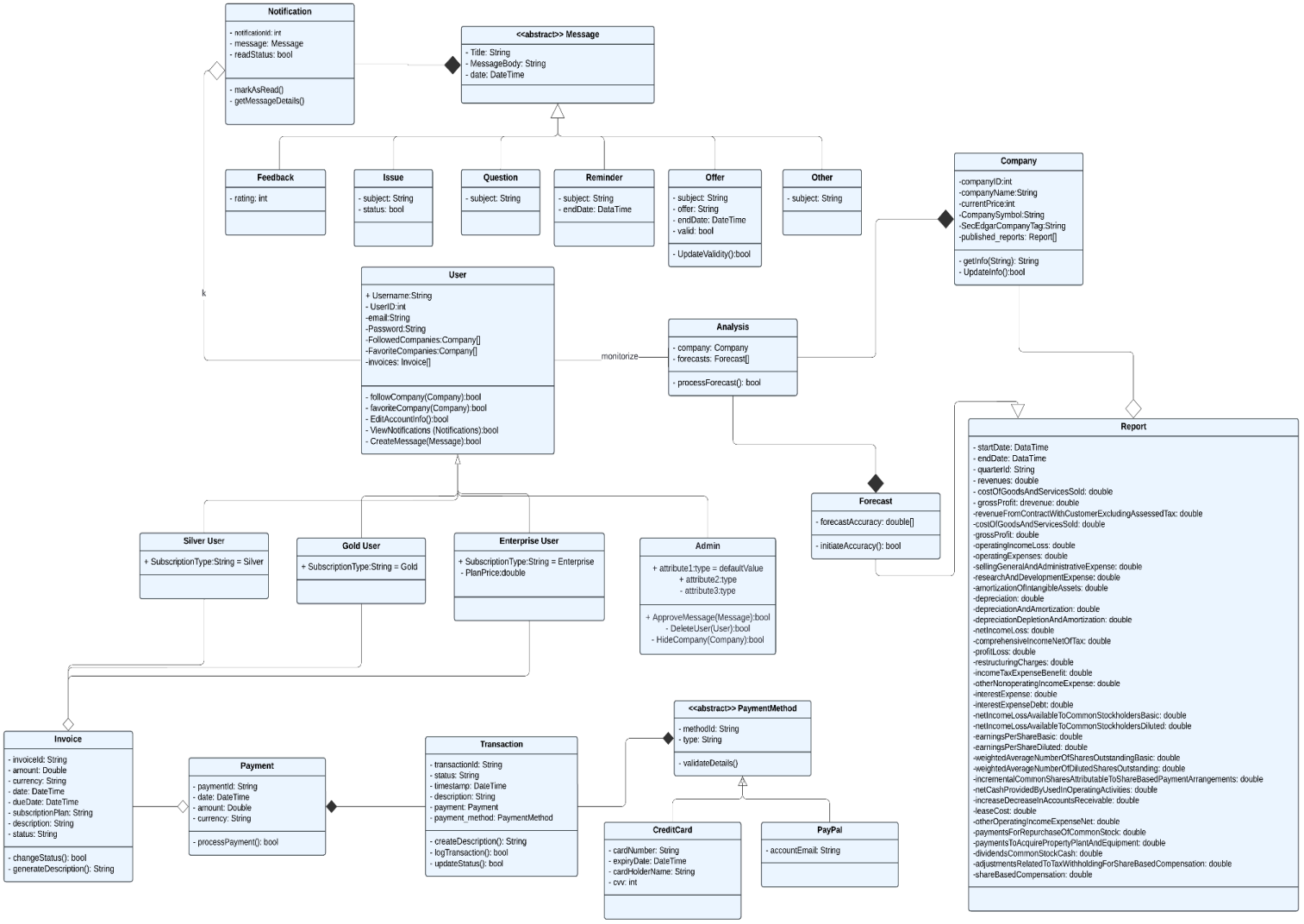
- **Issue cannot be resolved** → Administrator adds comments and notifies the user.

3.5.2. Use Case Model



The following use-case diagram is provided to summarize the functionalities and interactions within the system. While use-case diagrams are associated with systems designed using object-oriented programming paradigms, it is important to note that this project does not strictly follow an OOP structure. The diagram serves as a high-level representation of system behavior and user interactions but does not necessarily reflect an object-oriented implementation.

3.5.3. Object and Class Model



The User class is a central entity in this system. It manages user information through attributes like Username, UserID, email, and Password. Users can interact with companies via the followCompany() and favoriteCompany() methods and perform tasks like editing account information or viewing notifications. The User class has 4 specialized subclasses:

- Silver User: A user with a SubscriptionType set to "Silver".
- Gold User: A premium user with SubscriptionType set to "Gold".
- Enterprise User: This subclass includes an additional attribute, PlanPrice, to manage enterprise-level subscriptions.
- The Admin class represents system administrators who have control over the platform. They can approve or moderate messages with ApproveMessage(), remove users via DeleteUser(), hide companies using HideCompany().
- Users can monitor analysis and interact with notifications. Admins can moderate Users, Messages, and Companies.

The Notification class is responsible for managing alerts sent to users. Each notification has a message, which links to an abstract Message class. There are 5 types of Message:

- Feedback: Includes a rating attribute.

- Issue: Contains a subject and status to track problems.
- Question: Represents a query with a subject.
- Reminder: Sends reminders with a subject and an endDate.
- Offer: Contains promotional information with offer, endDate, and validity management through the UpdateValidity() method.
- Other: Represents generic message types

The Company class manages details about companies, including companyName, currentPrice, and CompanySymbol. It has a published_reports array that represents the published quarter reports of that company. published_reports array's type is Report, a class containing extensive financial metrics, such as revenues, grossProfit, operatingExpenses, taxes, and other financial parameters. To be more concise, the Report class keeps the required quarter report information gathered from SecEdgar API.

The Company class is also closely linked with Analysis, which processes forecasts. The Forecast class represents the forecasted quarter report by the system. It calculates forecastAccuracy and has the initiateAccuracy() method to validate these results.

The Invoice class manages financial bills, containing attributes like invoiceId, amount, currency, and status. It supports methods such as: changeStatus(): Updates the current invoice status (Paid, Unpaid, Overdue). generateDescription(): Generates descriptive content for invoices.

Invoices link to the Payment class, which processes payments with a paymentId, amount, and currency. Here, you can think of invoice as a debt and payment as the payment of this debt. Payments rely on PaymentMethod, an abstract class. This abstraction supports two specific methods:

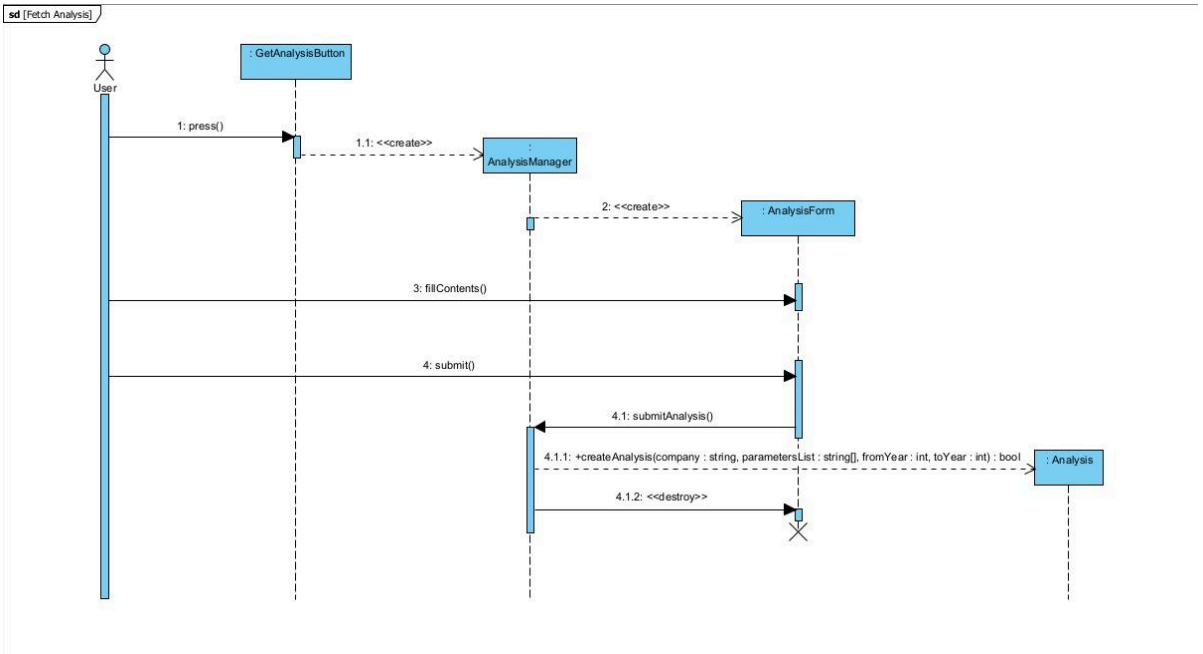
- CreditCard: Handles card-based payments with details like cardNumber, expiryDate, and cvc.
- PayPal: Manages payments via accountEmail

The Transaction class captures financial exchanges in the system. Attributes include transactionId, amount, timestamp, and a description. Transactions utilize PaymentMethod for processing. Methods like createDescription() and updateStatus() ensure transaction tracking and management.

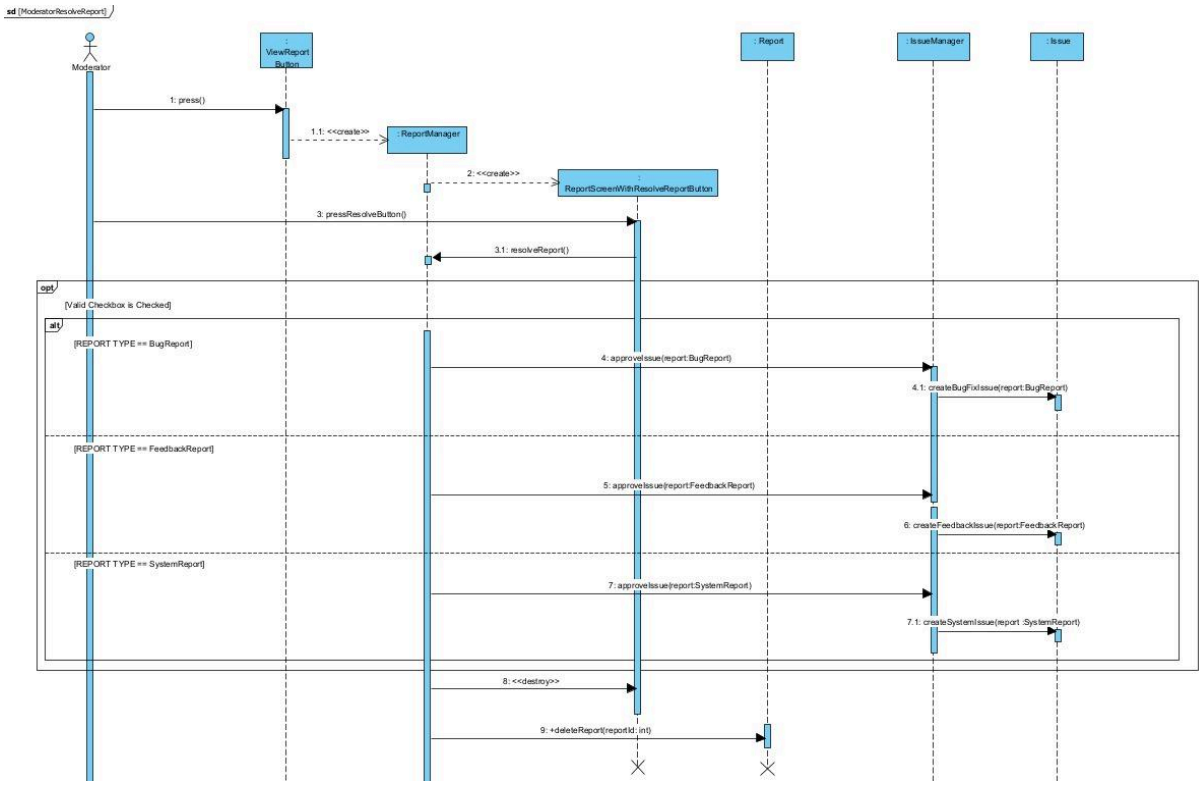
3.5.4. Dynamic Models

Sequence Diagrams:

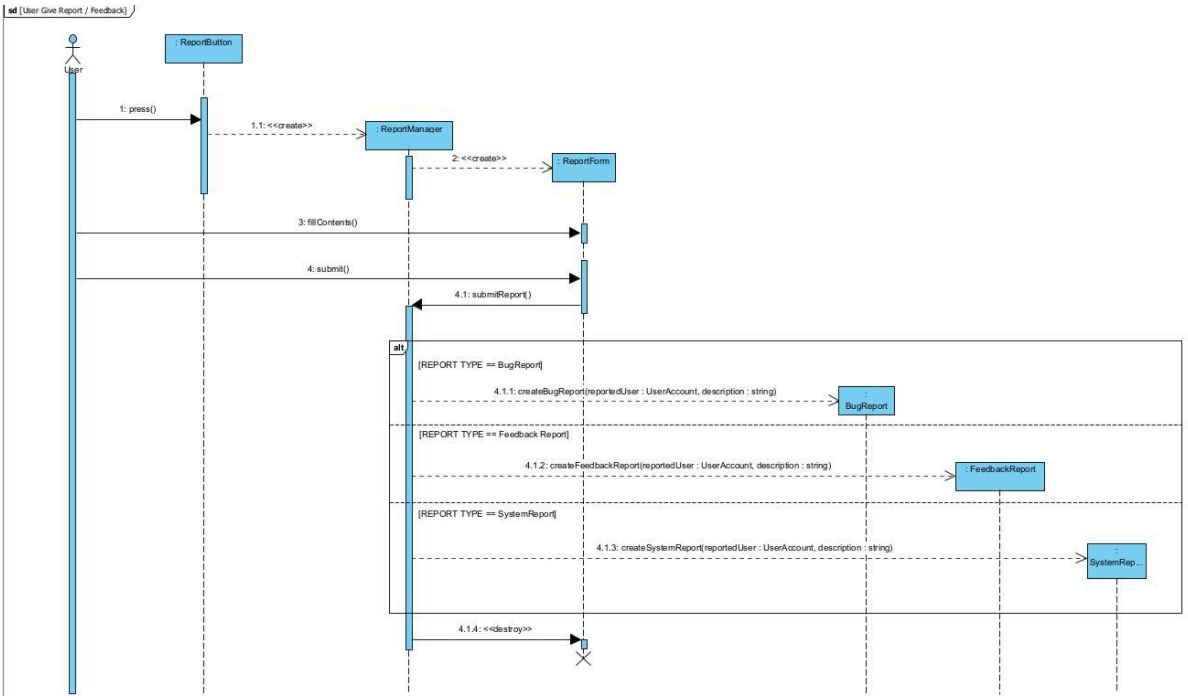
Fetching company specific analysis



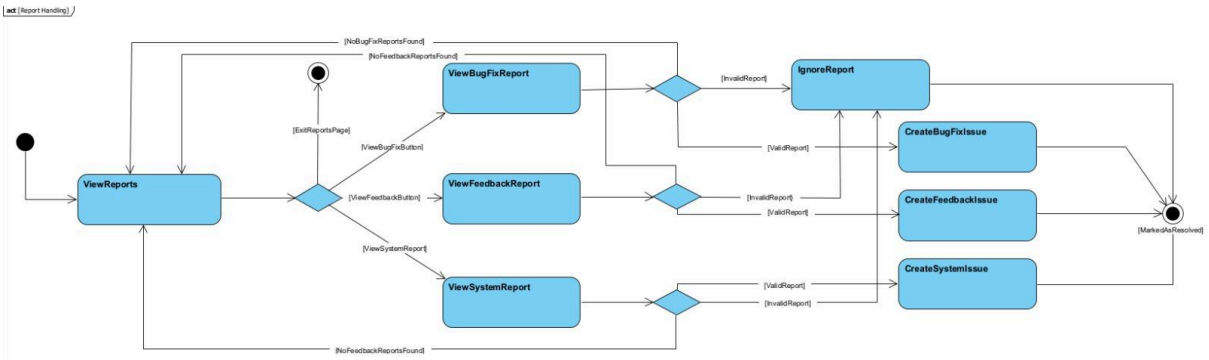
Moderator resolve reports



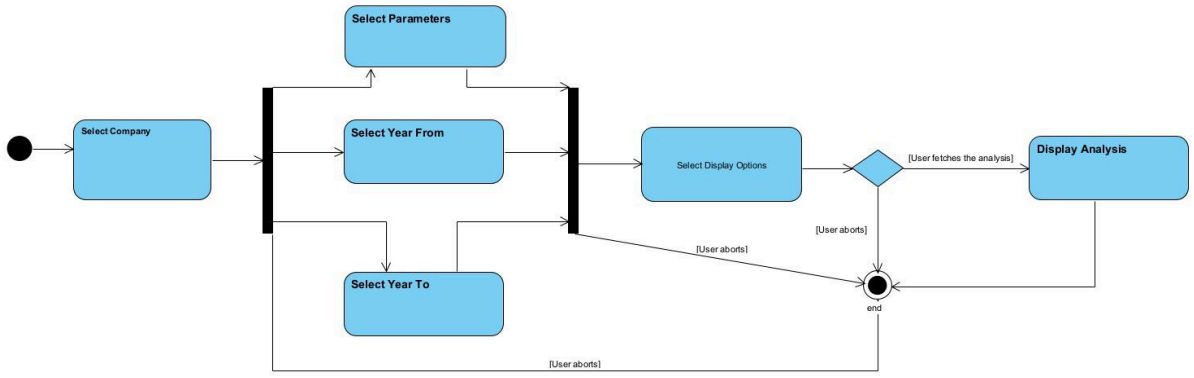
User submit report



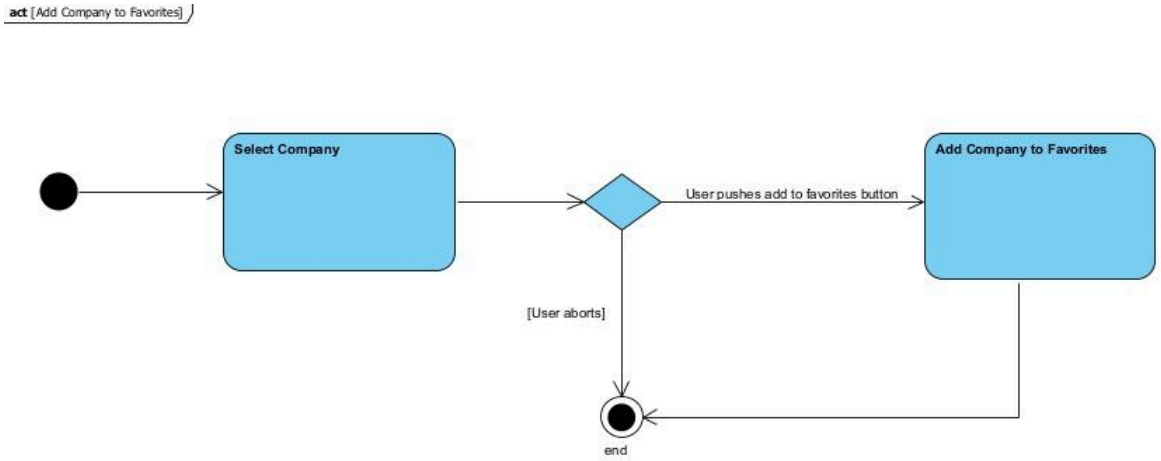
Activity Diagrams:
Moderator handles report



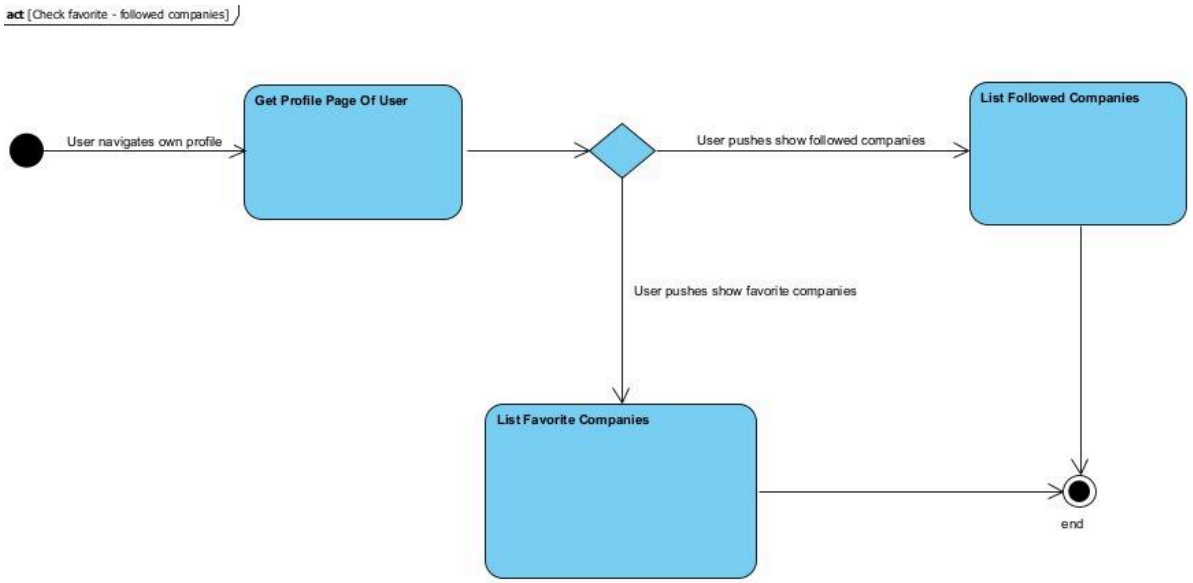
User get company specific analysis



User add company to favorites

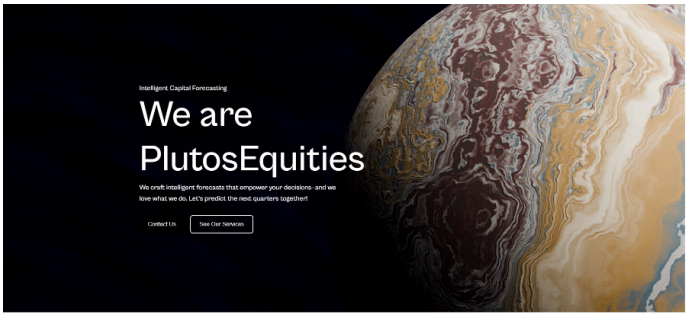


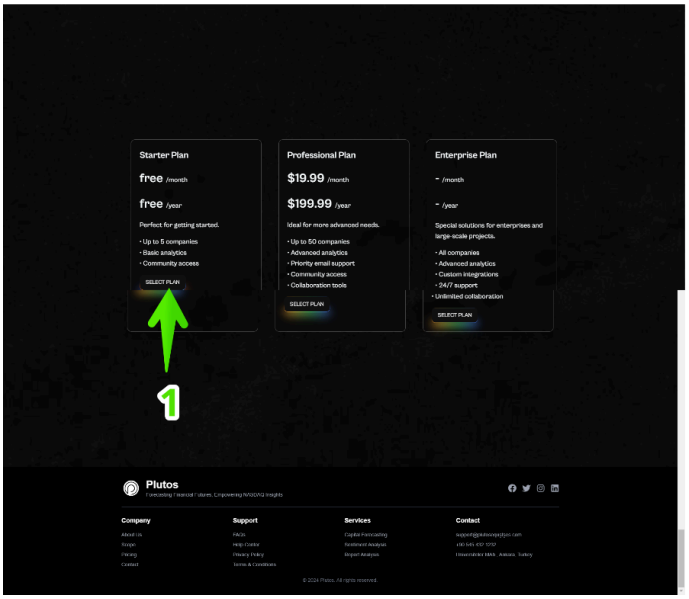
User check favorite and followed companies



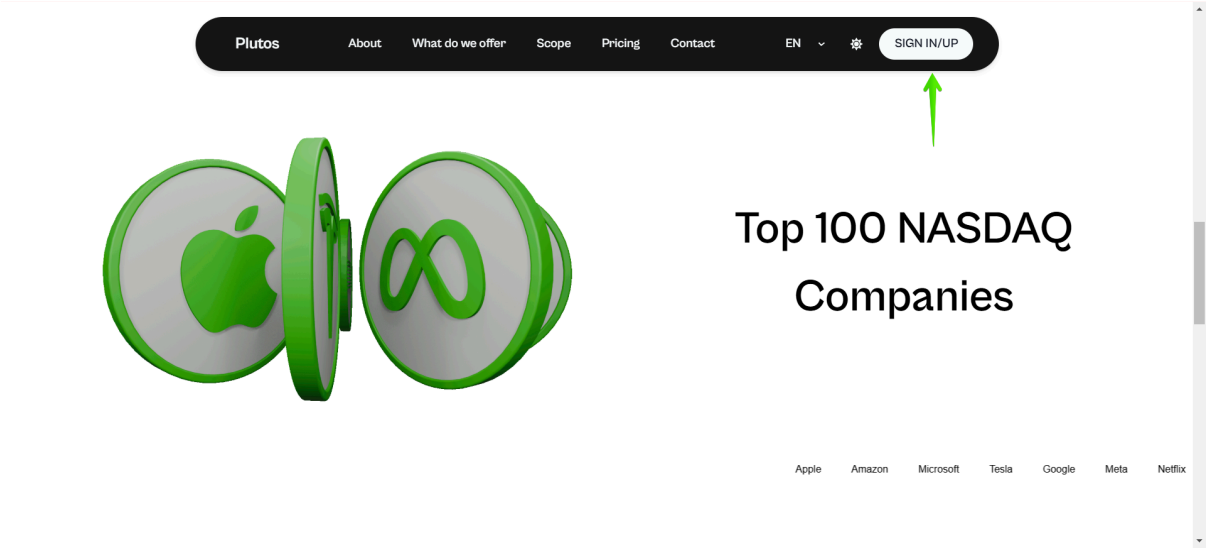
3.5.5. User Interface - Navigational Paths and Screen Mock-ups

Landing



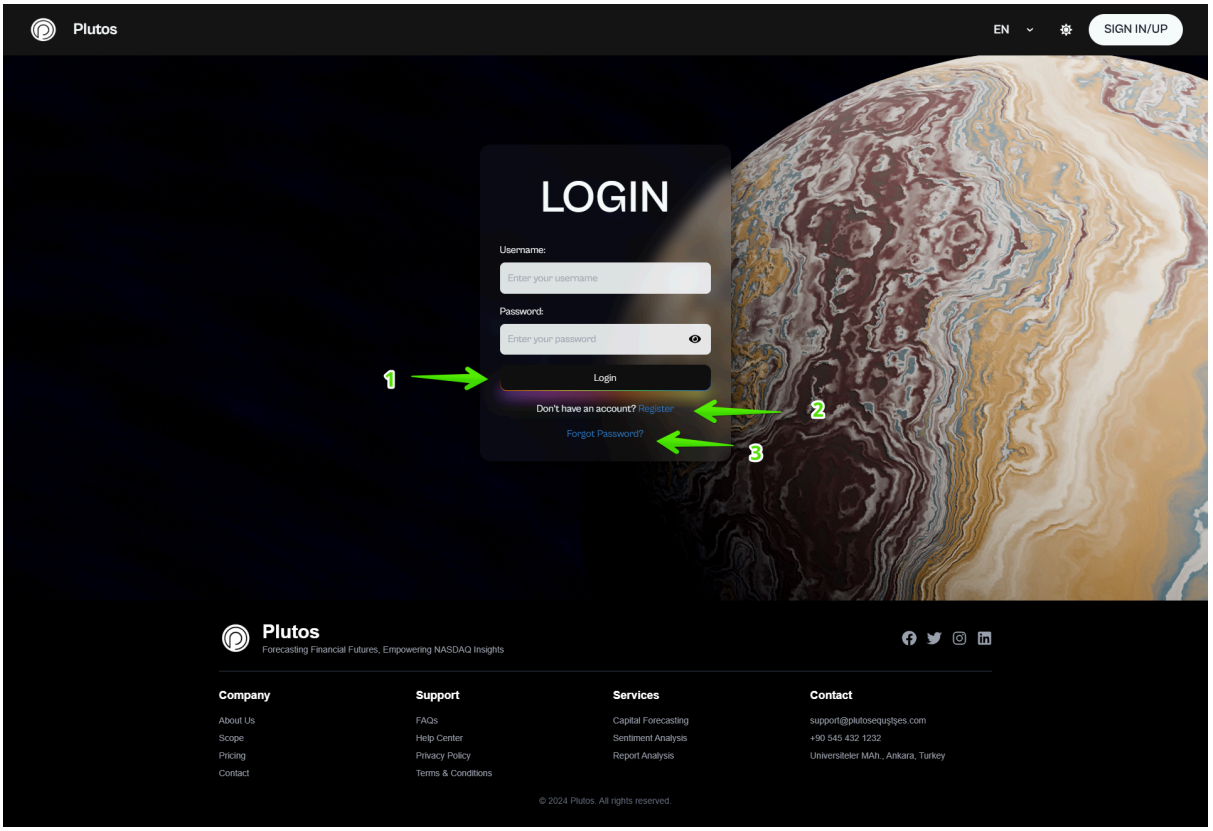


1. On the landing page, users can interact only with the **navigation bar** or the **Select Plan** buttons.
2. By clicking on Select Plan (press 1) in the landing page you will be directed to the **Registration** page.

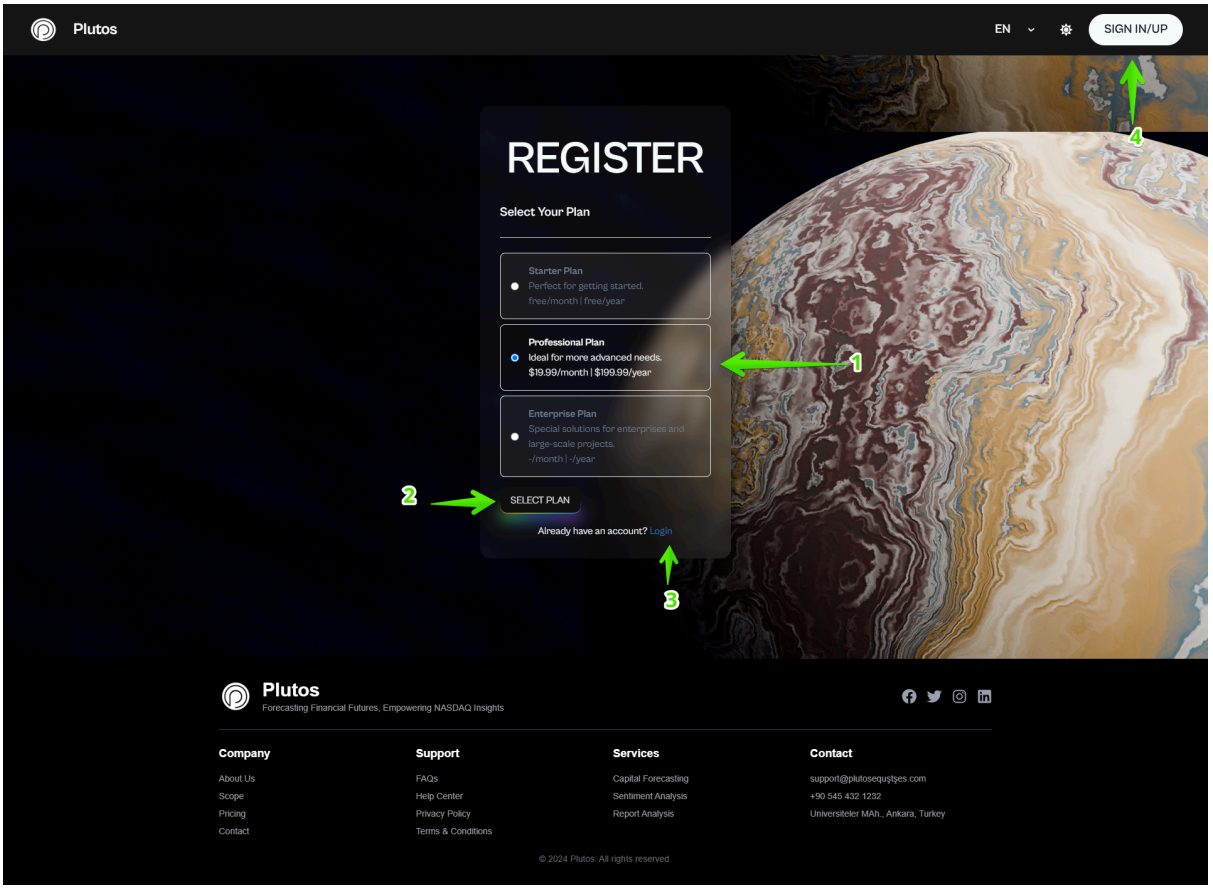


1. Green arrow shows the **navigation bar** on the landing page. Here, “About, What do we offer, Scope, Pricing and Contact” are directing users to different parts of the landing page. I.e. Currently the user is on the **Scope** section of the landing page.
2. “**EN**” in the **navigation bar** indicates English and if the user clicks on it it can change to Turkish (a dropdown box will be shown to let the user choose the language).
3. The **sun icon** represents theme selection. By clicking on it, the user can change its light theme to dark theme.
4. By clicking on the **Sign In/Up** button which the arrow is pointing at, the user can continue with its authentication.

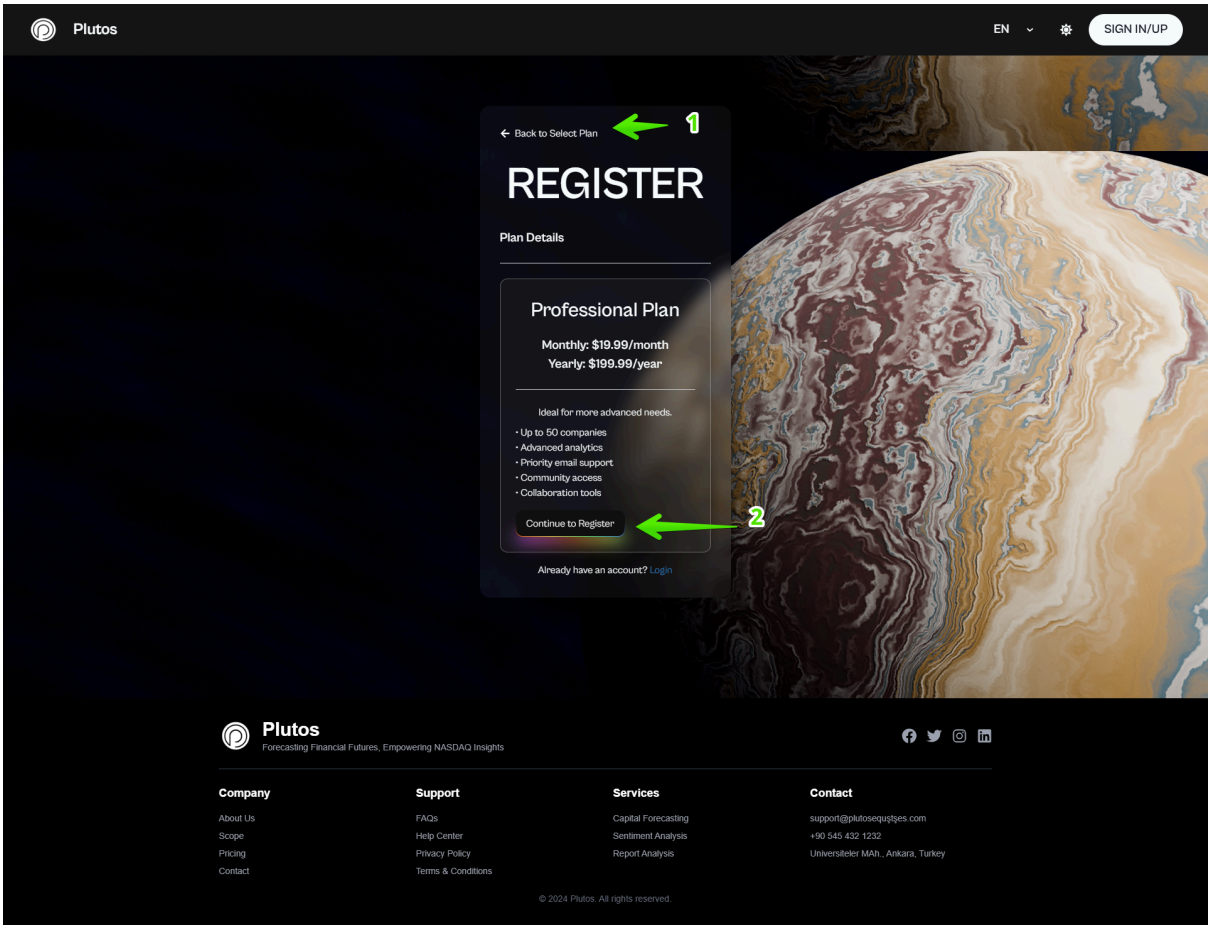
Authentication



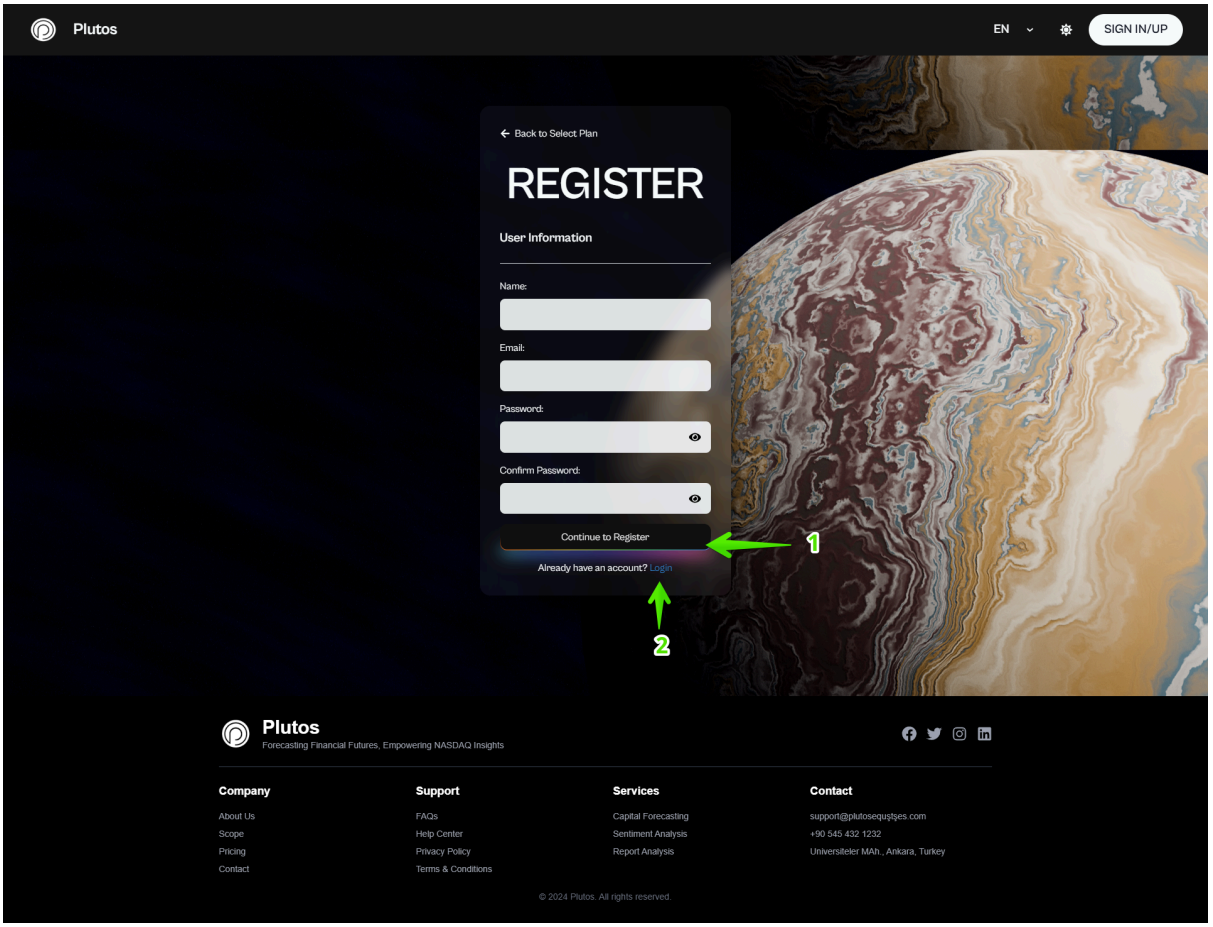
1. Login after entering credentials. (press 1)
2. Register to create an account (if does not exist). (press 2)
3. Reset your password in case of forgetting it. (press 3)



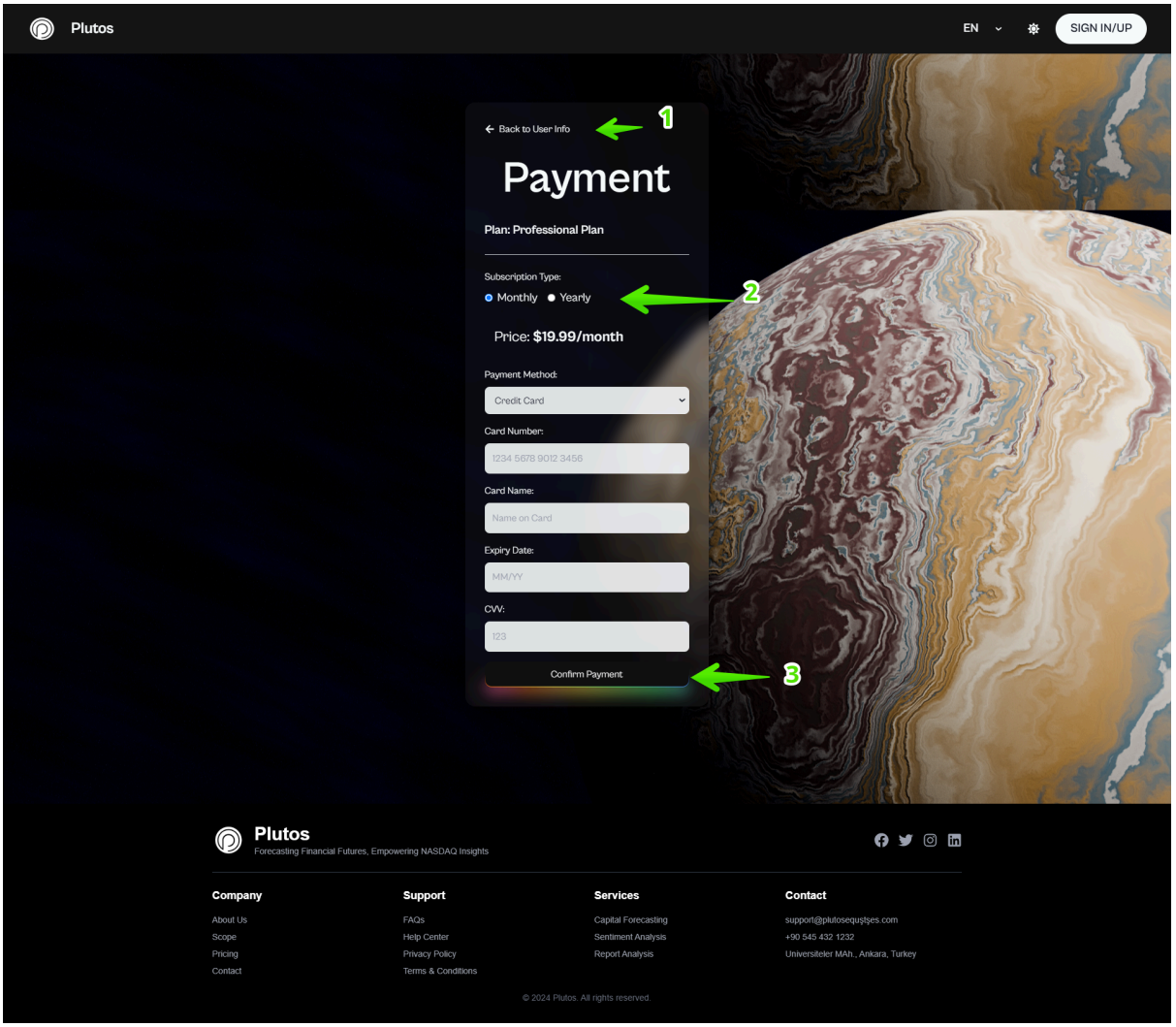
1. Select your plan among 3 different options. (i.e. press 1)
2. Continue the registration process. (press 2)
3. Go to login page. (press 3)
4. Alternatively, go to the login page by pressing the Sign In/Up button. (press 4)



1. Go back to selecting plans to change your plan. (press 1)
2. Continue with registration. (press 2)

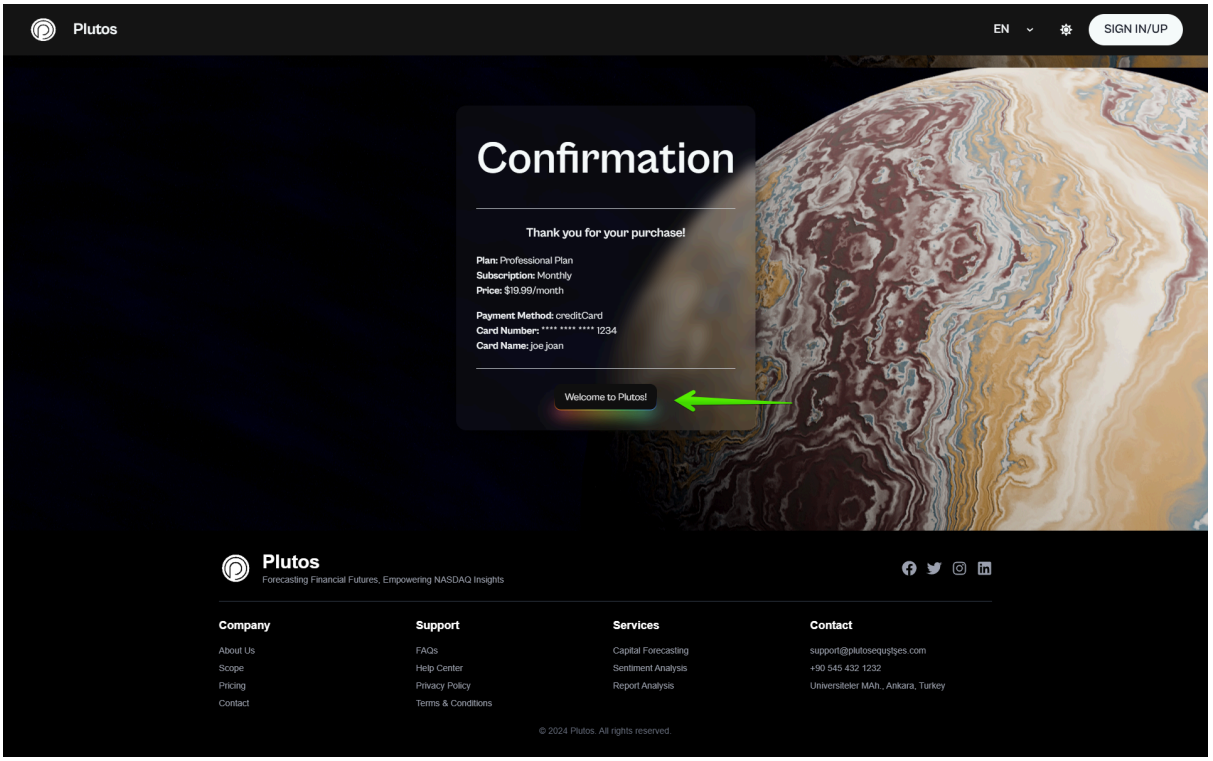


1. After entering your information continue with registration. (press 1)
2. If you already have an account login with that account. (press 2)



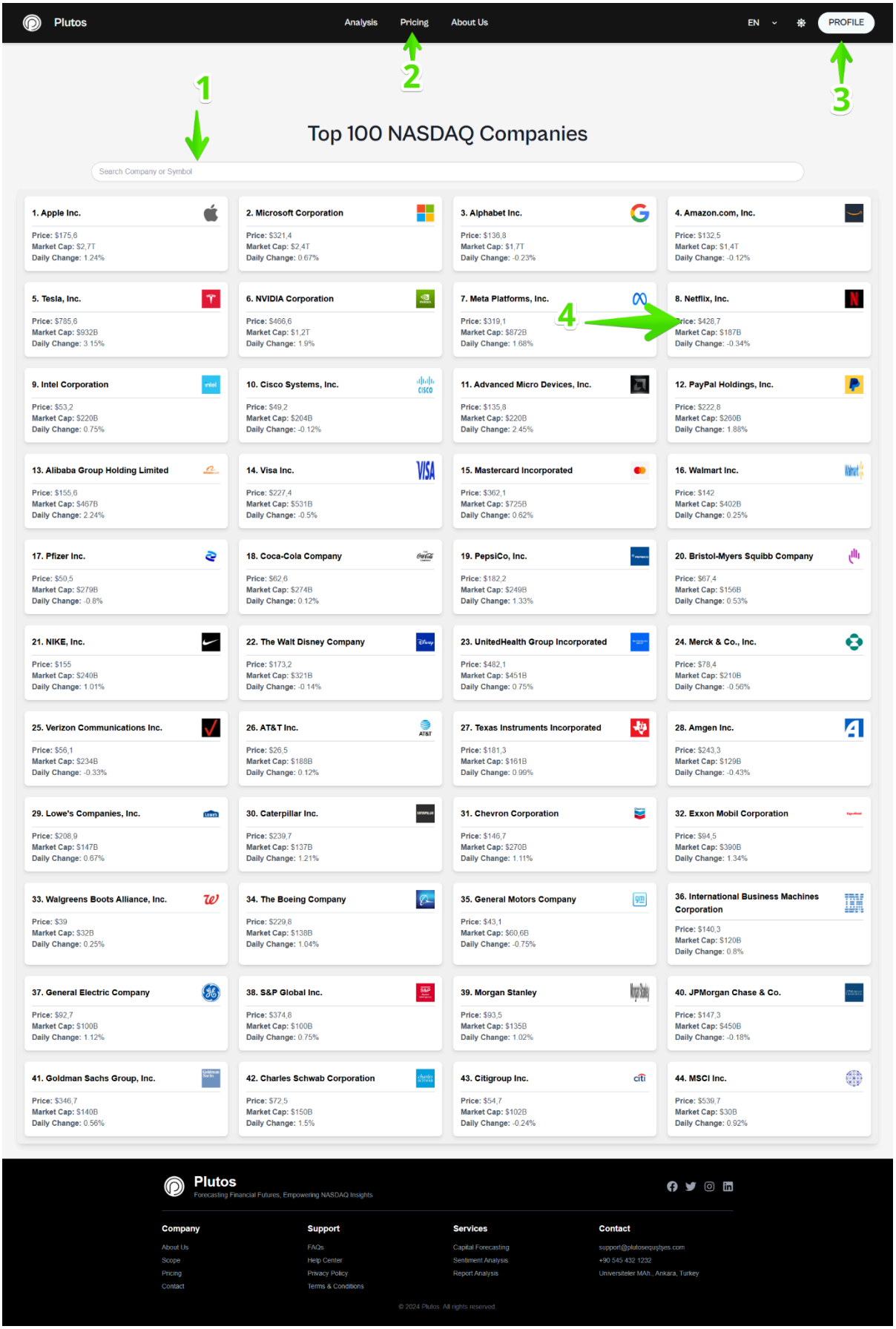
1. Back to the user information page to change them. (press 1)
2. Select your period/duration of subscription. (press 2)
3. After selecting your payment method, fill in necessary information for payment and complete payment. (press 3)

Note: If you choose a free plan this payment page won't be not shown and you will be directly registered.



1. Continue with the main page after confirmation of payment

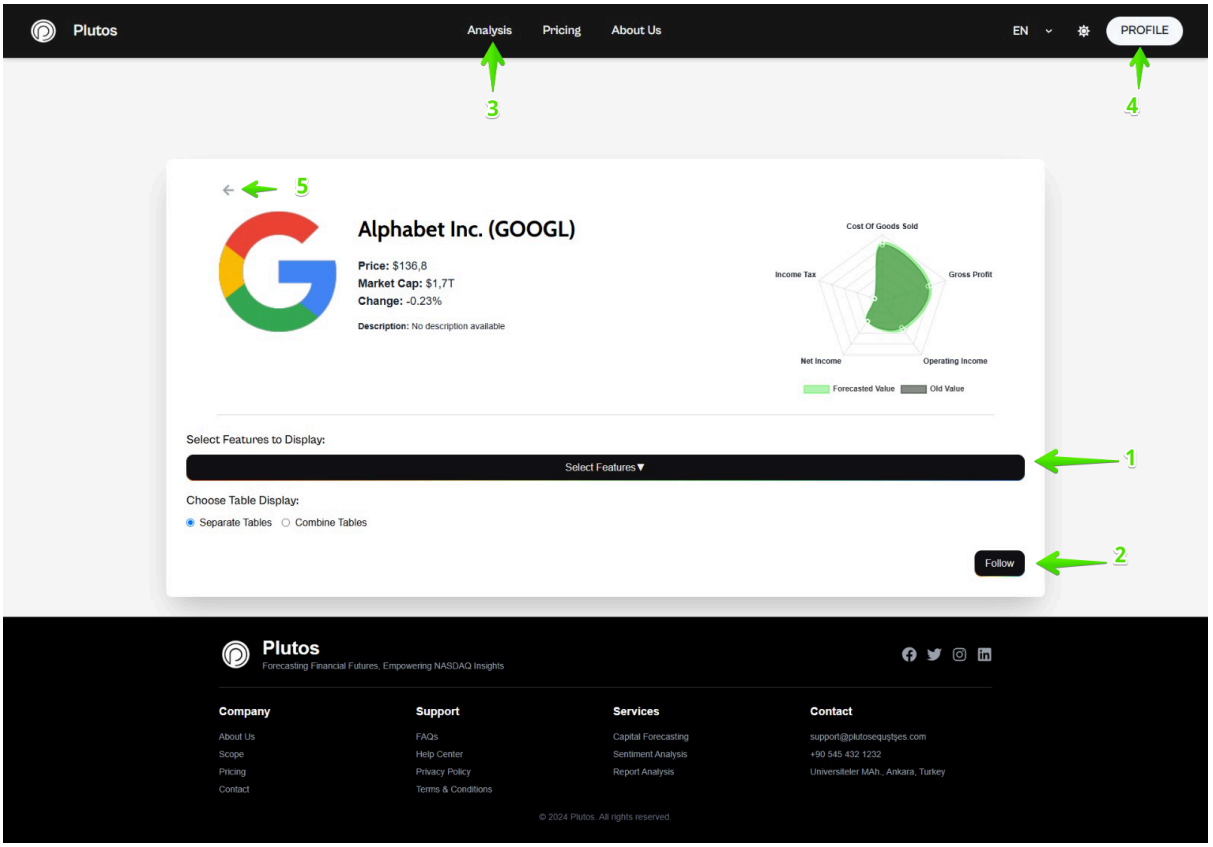
Home/Main Page



Here we see the main/home page that the user is directed after the authentication. Here, you can:

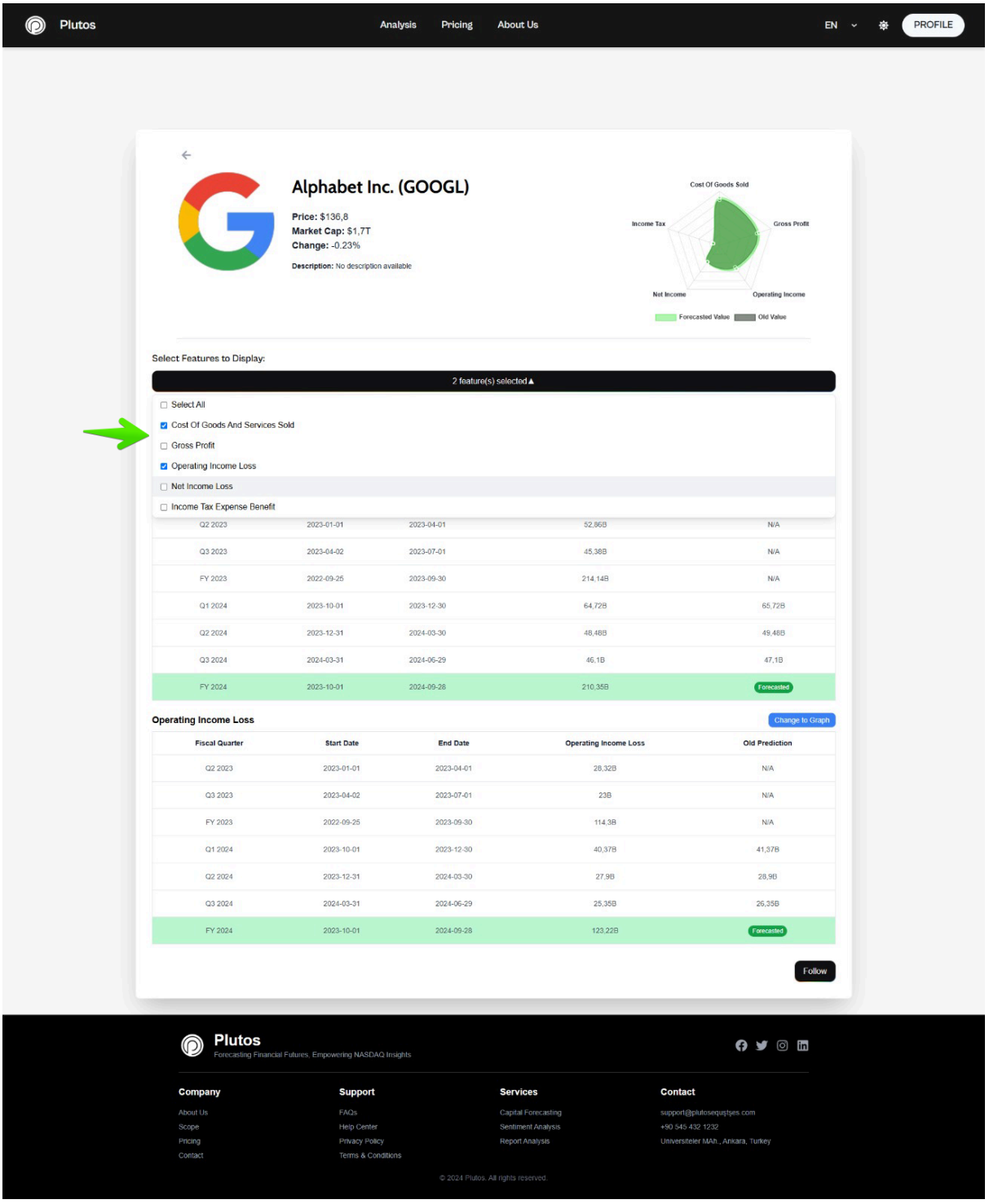
1. Search company by name or symbol/ticker. (press 1)
2. On the navigation, you see 3 button options (see 2):
    a. Analyze button to navigate to the main/home page. (current page).
    b. Pricing button to navigate to the subscription section located in the profile.
    c. About Us button to navigate a simple page that displays a brief descriptive text about us.
3. Go to your own profile page. (press 3)
4. Go to the analyse page of companies. To do that you simply have to click on the company widget. (e.g. to see Netflix analysis, press 4)

Analysis Page

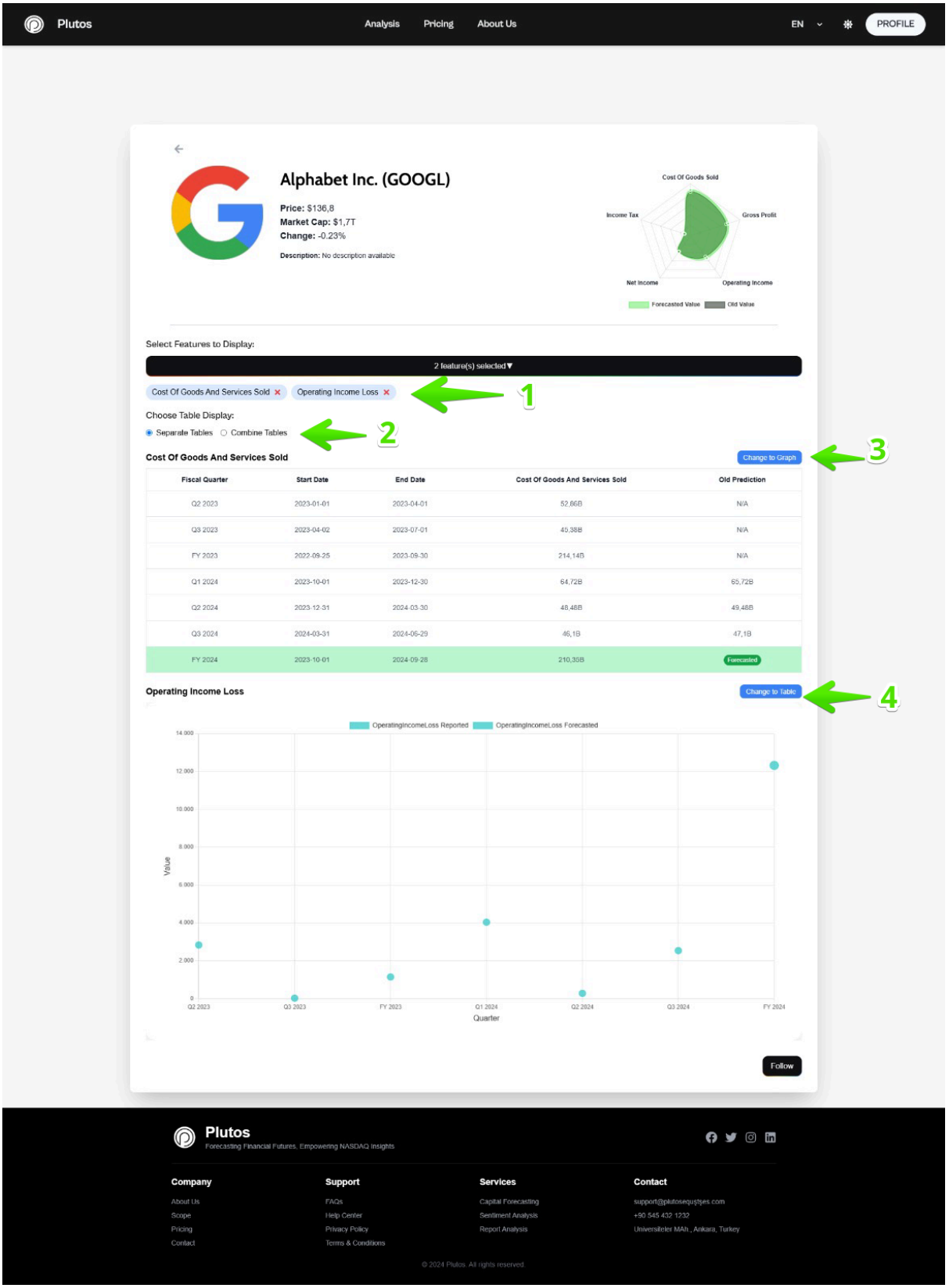


Let's assume you have clicked on Google widget and it opens Google's analysis page. Here you can:

1. Select Features you want to see forecast and history for. (press 1)
2. Press the Follow button to save the companies you want to follow. You see it on your profile dashboard. (press 2)
3. Go back to the home/main page where companies are listed. (press 3 or press 5)
4. Go to your Profile page. (press 4)

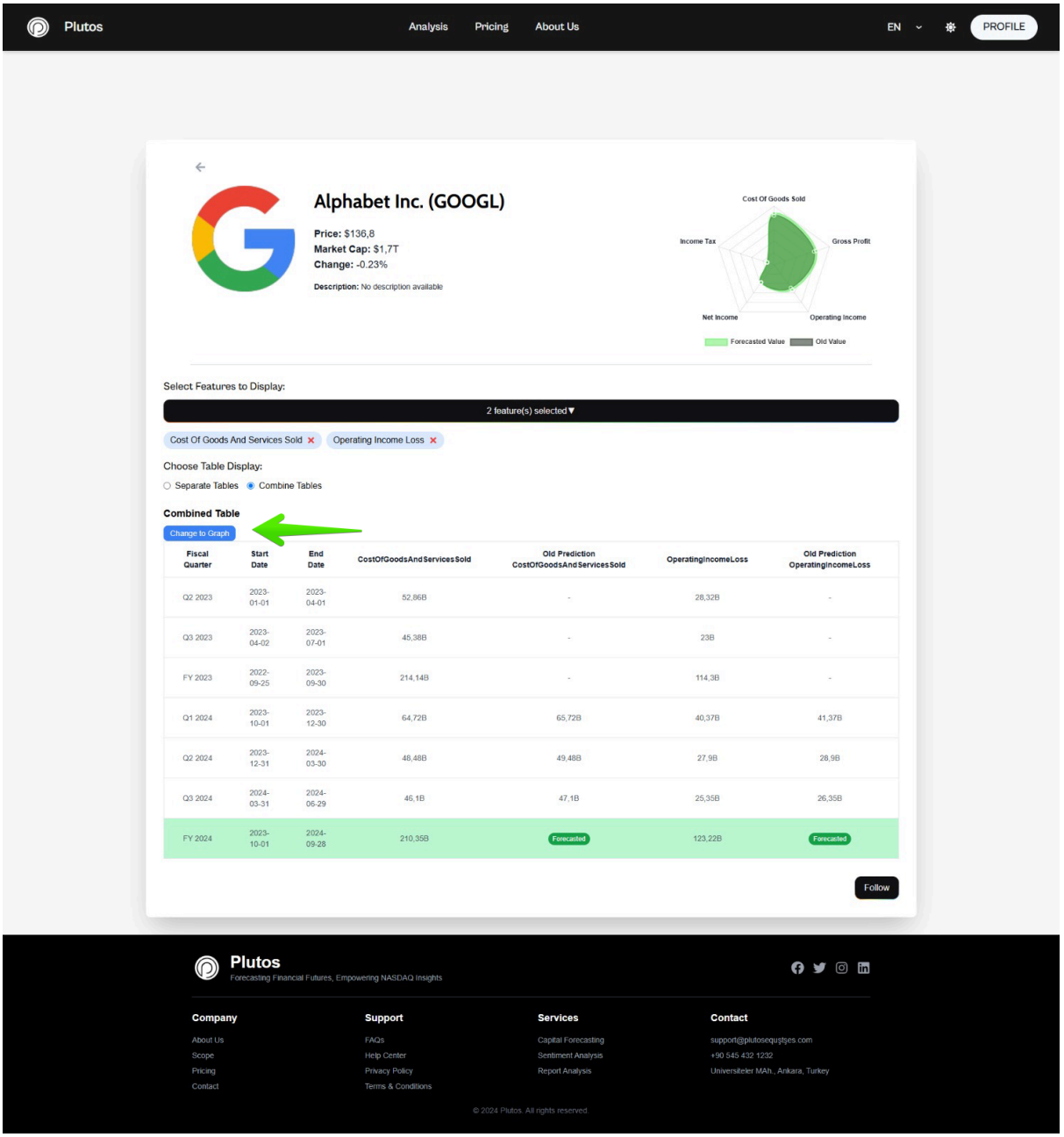


1. After you press the features dropdown box, you can select the forecasted features that you want to see.

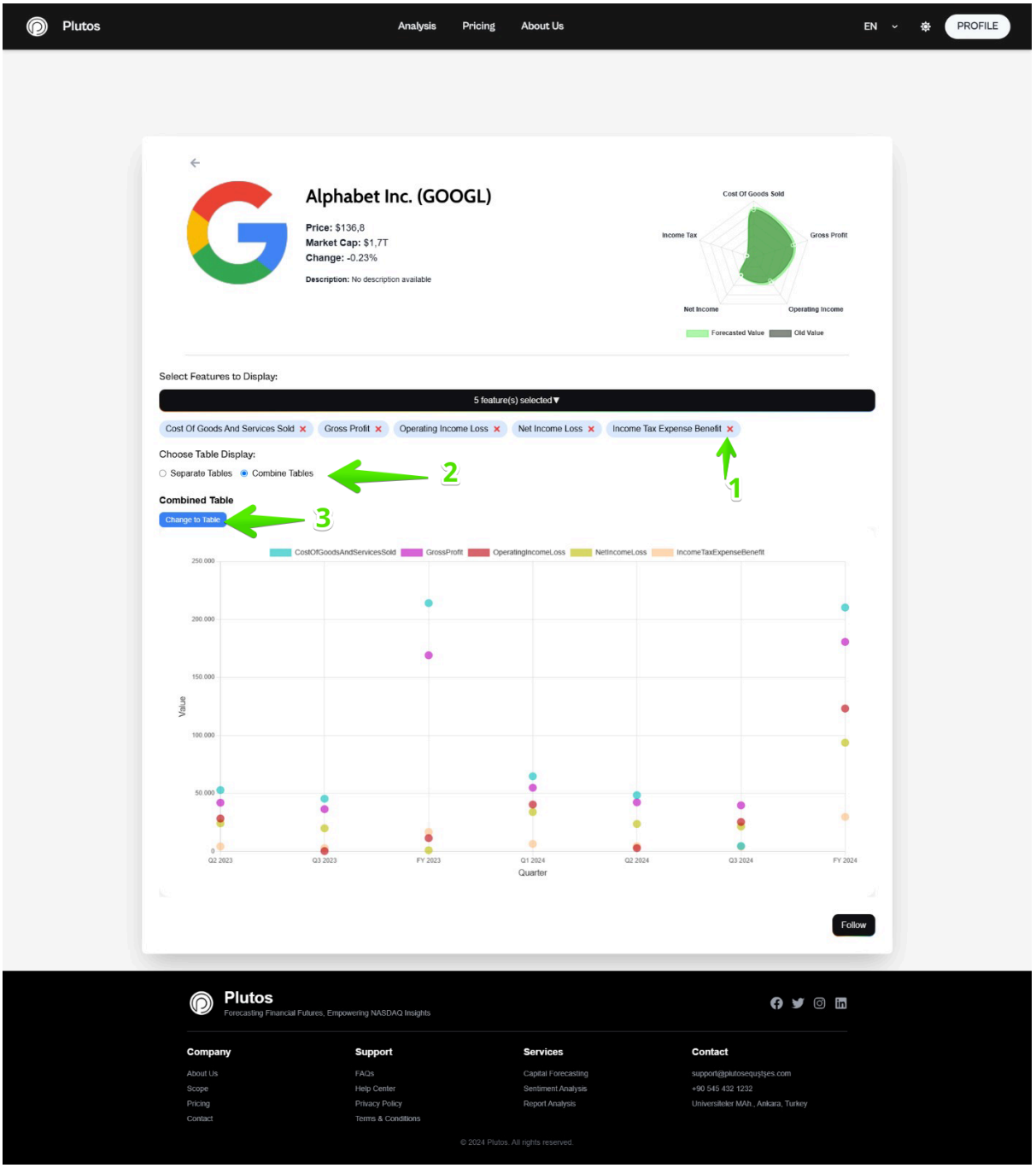


After choosing the features. You can:

1. Unselect the parameters that you don't want to see. (press 1)
2. Merge/divide the individual tables of different parameters. (press 2)
3. Press Change to Graph to visualize the table. (press 3)
4. Press Change to Table to monitor predictions in tabular format. (press 4)



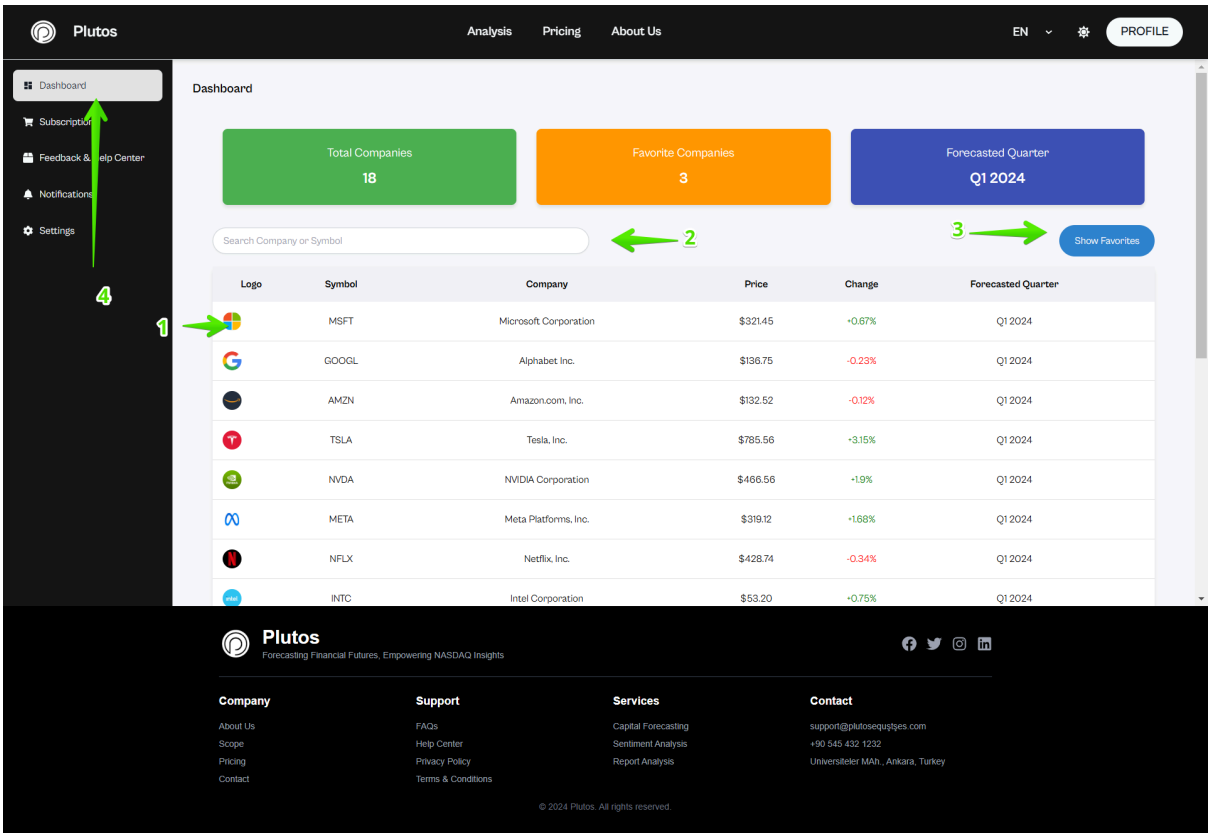
If you press Combine radio button, you will see this frame.You can turn this table into a graph by clicking on the Change to Table button as shown by the green arrow.



Let's assume you selected all features, clicked on the combine radio button, and press change to graph option. Here you can:

1. Remove a displayed feature. (press 1)
2. Separate table to see each feature individually. (press 2)
3. Change the graph to a table. (press 3)

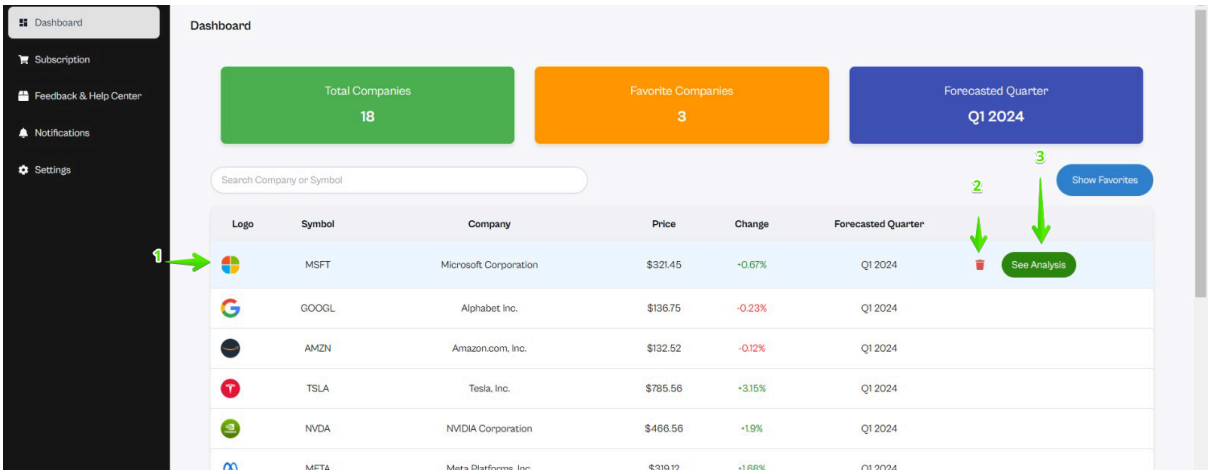
Profile Page



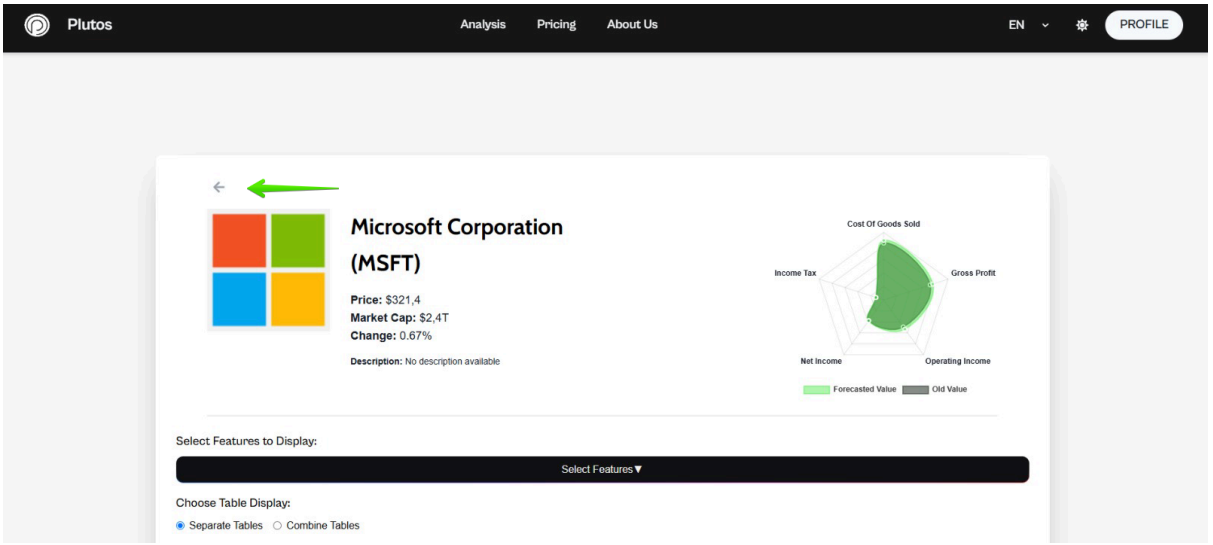
When you press the Profile button on the top corner, you will be directed to this page. In the profile, there are 5 sections (Dashboard, Subscription, Feedback & Help Center, Notification and Settings). By default, the dashboard section opens. Dashboard permits users to have quick access to a specific company and to see its analysis faster. On dashboard frame, you see:

1. List of followed companies from the Dashboard. (see 1)
2. Search a company by name or symbol/ticker. (press 2)
3. See your list of favorite companies by pressing the Show Favorites button. (press 3)
4. Button to navigate to the Dashboard. (see 4)

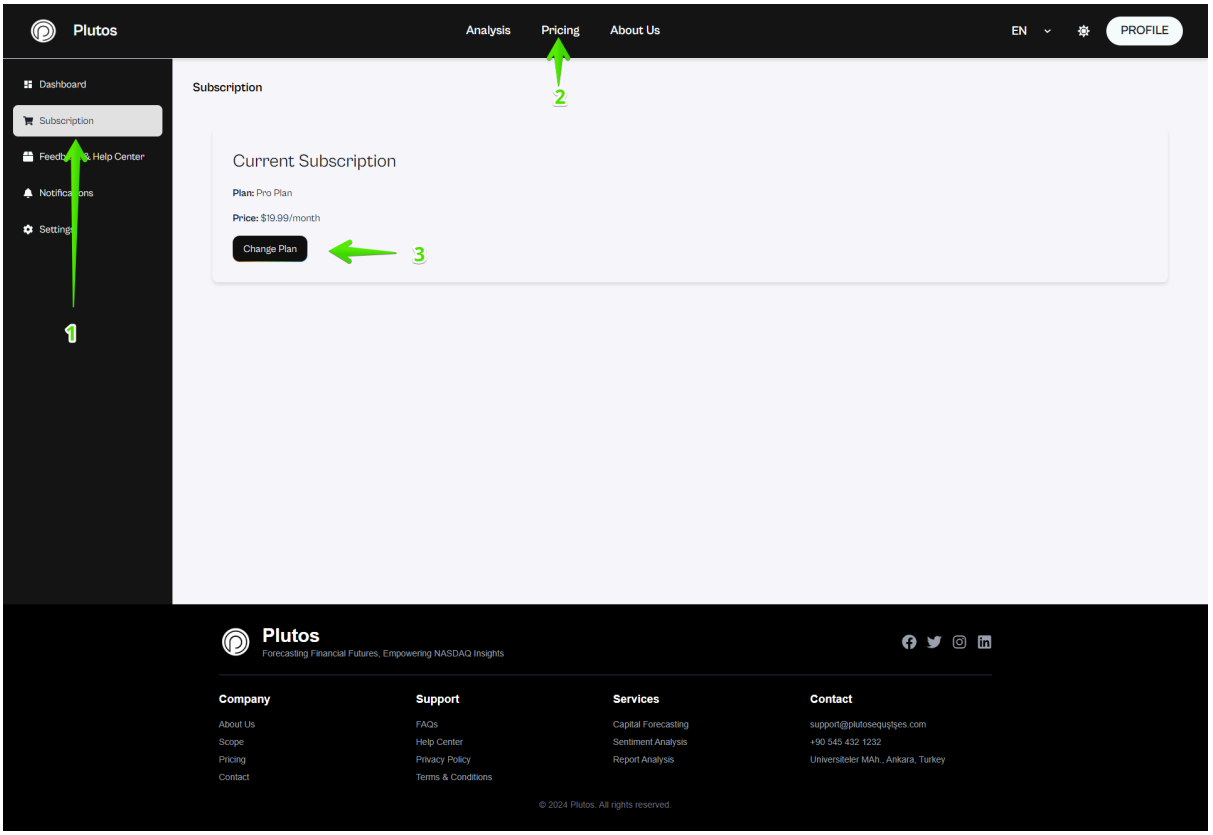
Note: Favorite companies are displayed in the exact layout and UI as the following companies.



1. When you click on a company row. 2 buttons appear: Unfollow button (trash icon) and See Analysis button. (press 1)
2. To unfollow a company, click on the trash icon. (press 2)
3. To see company analysis click on the See Analysis button. It navigates you to the Analysis Page. (press 3)

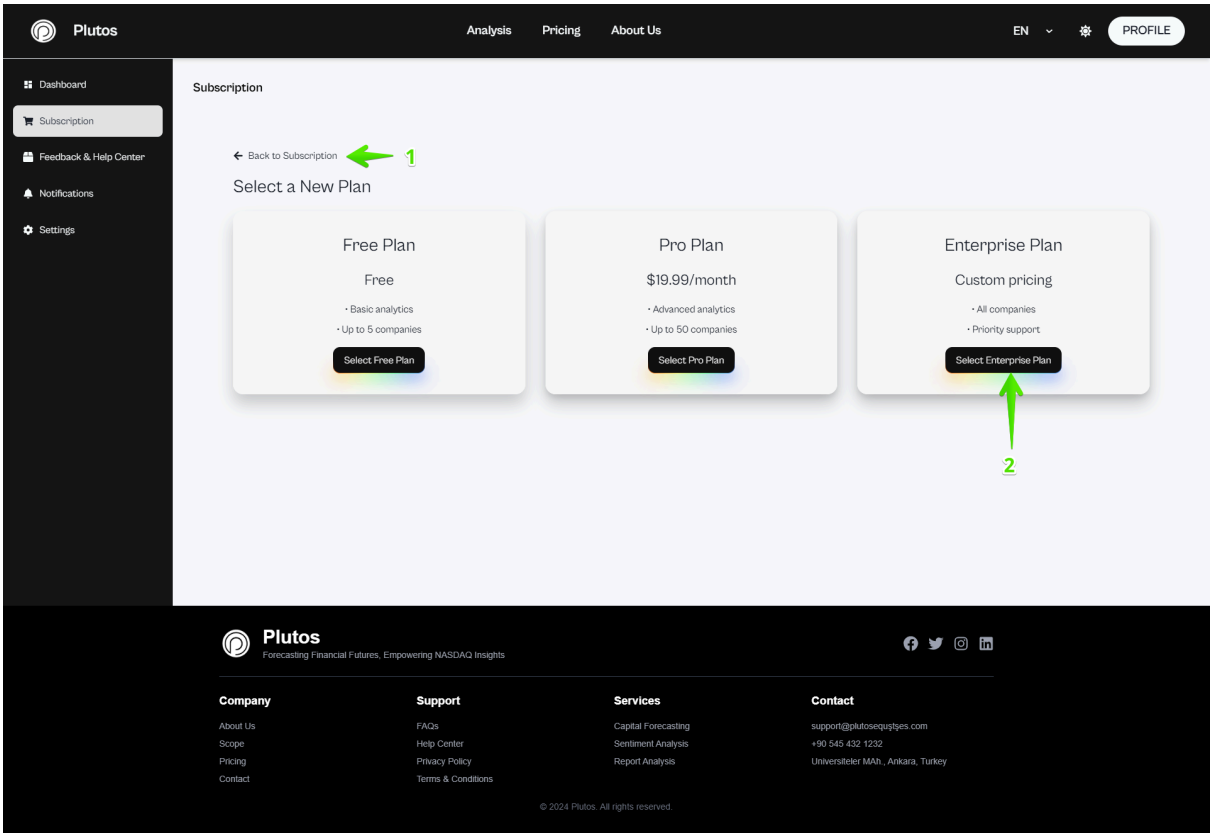


1. When you open the Analysis Page from your profile, you can go back to your profile by clicking on Back button shown by green arrow.



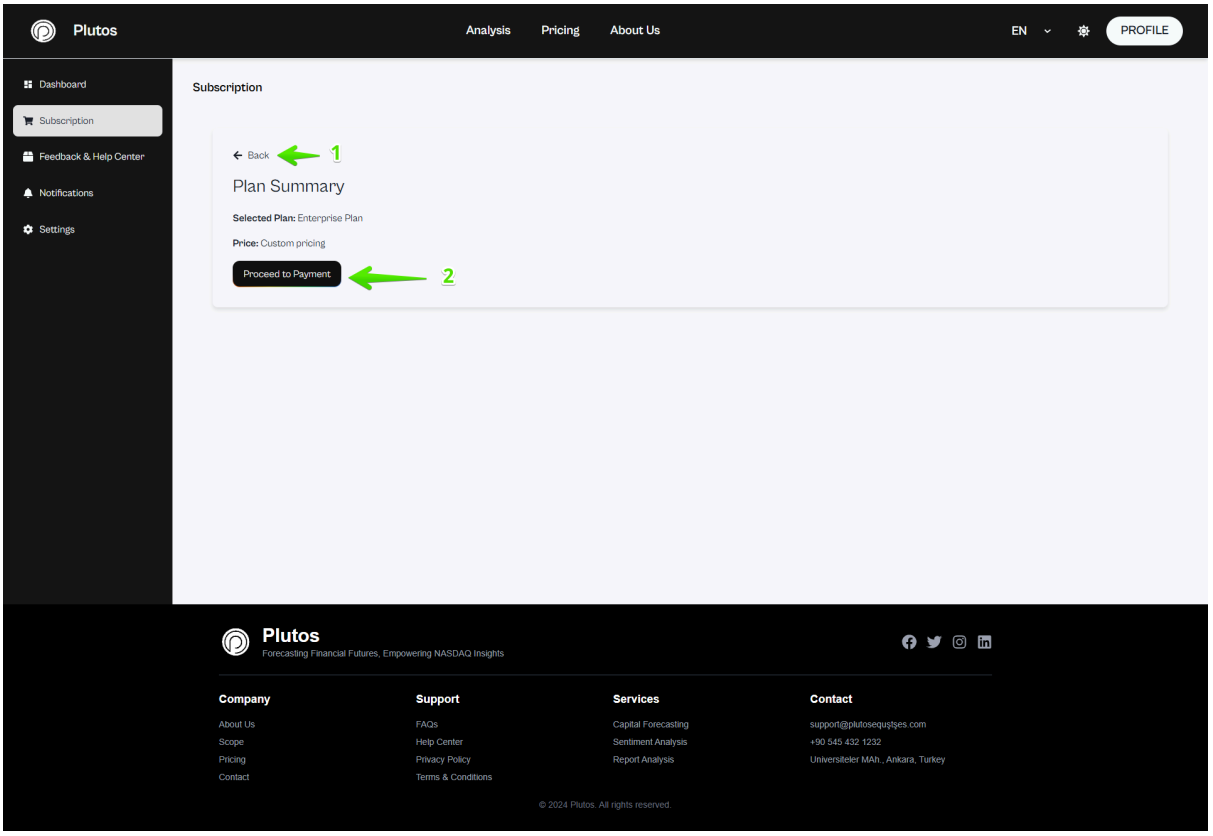
Second section of the profile is Subscription. This section can be opened from profile clicking on the Subscription option on the sidebar or Pricing option on the navigation bar. (press 1 or 2)

1. Click to see your current plan. (press 3)



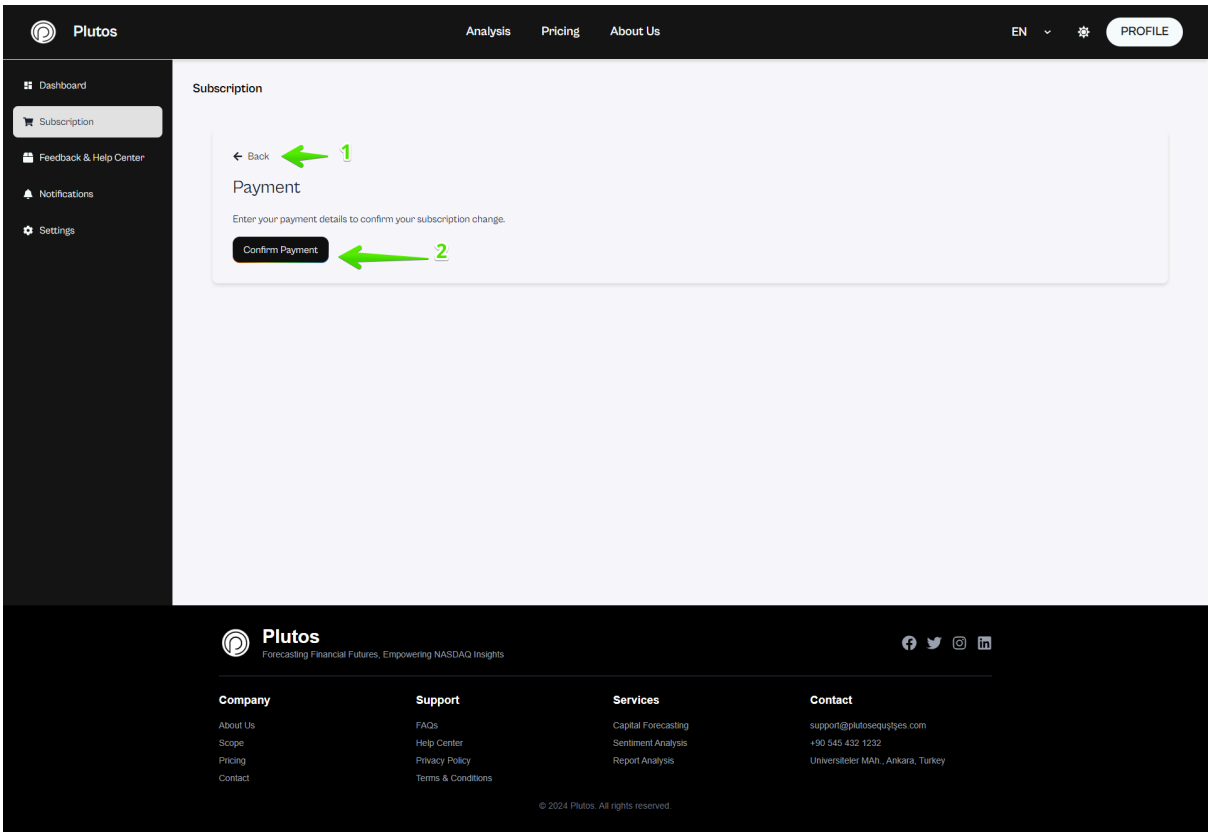
After clicking on the Change Plan button, Select New Plan frame is shown. Here you can:

1. Go back by pressing the Back to Subscription button. (press 1)
2. Continue by selecting and clicking on change your current plan button. (press 2)

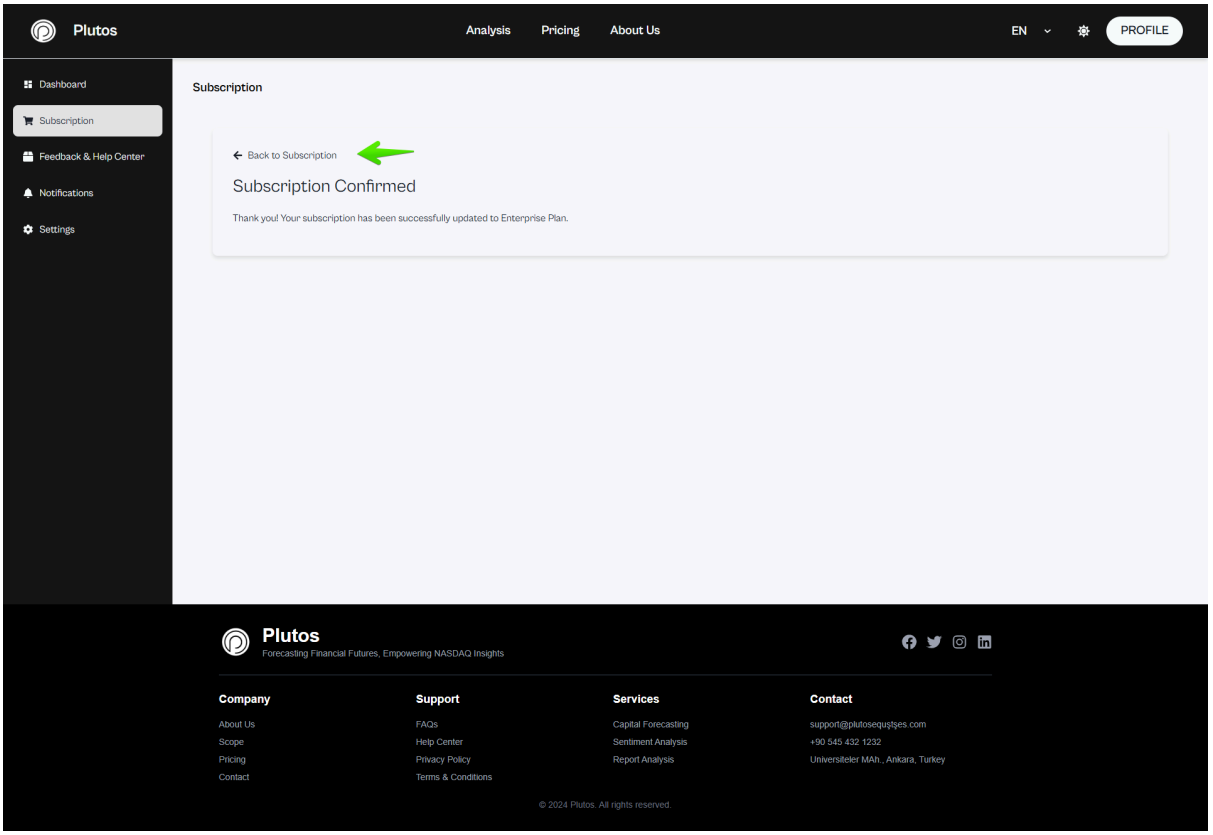


After selecting a plan, you see its summary displaying the plan's name and its price. You can:

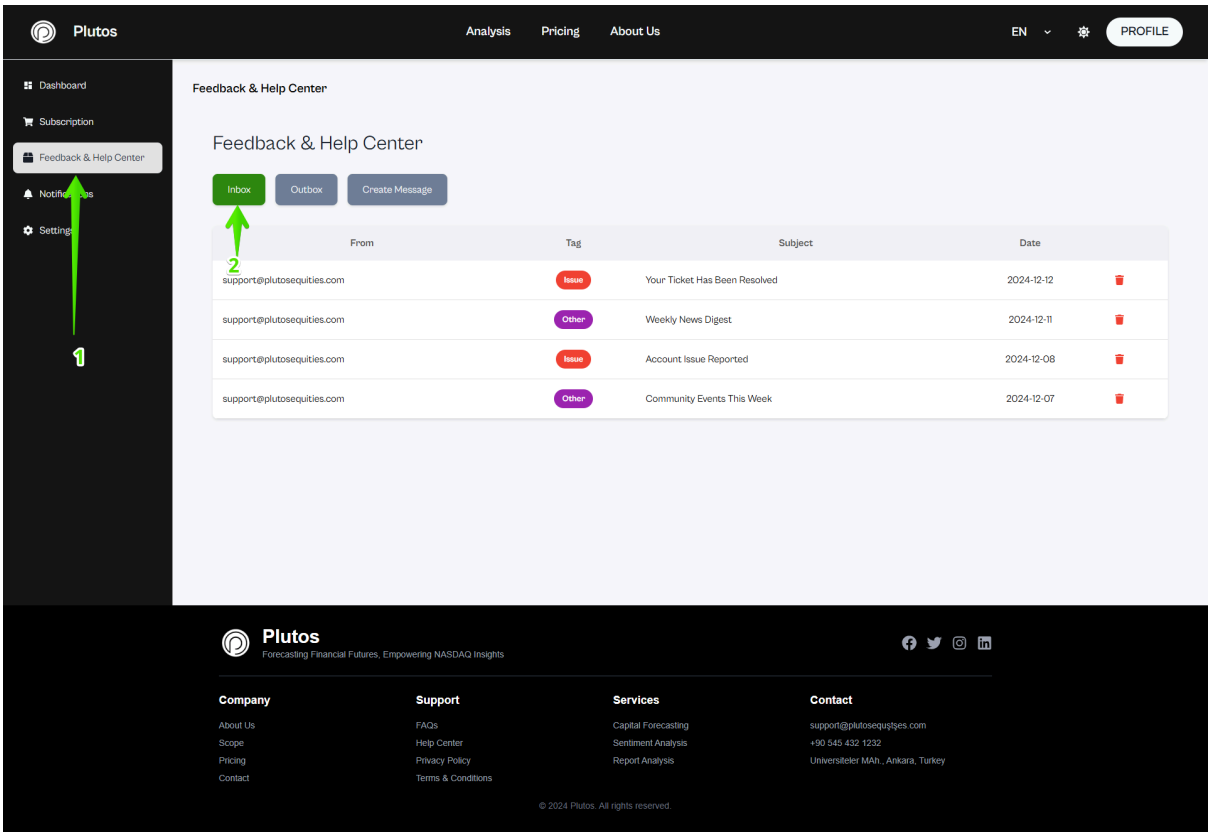
1. Go back by pressing Back button. (press 1)
2. Continue by pressing Proceed to Payment. (press 2)



1. In this page, you enter the payment details just like in the registration. However, if you choose a free plan you don't have to enter payment credentials. Then, click on Confirm Payment to change your plan. (press 2)
2. You can still go back by pressing the Back button. (press 1)



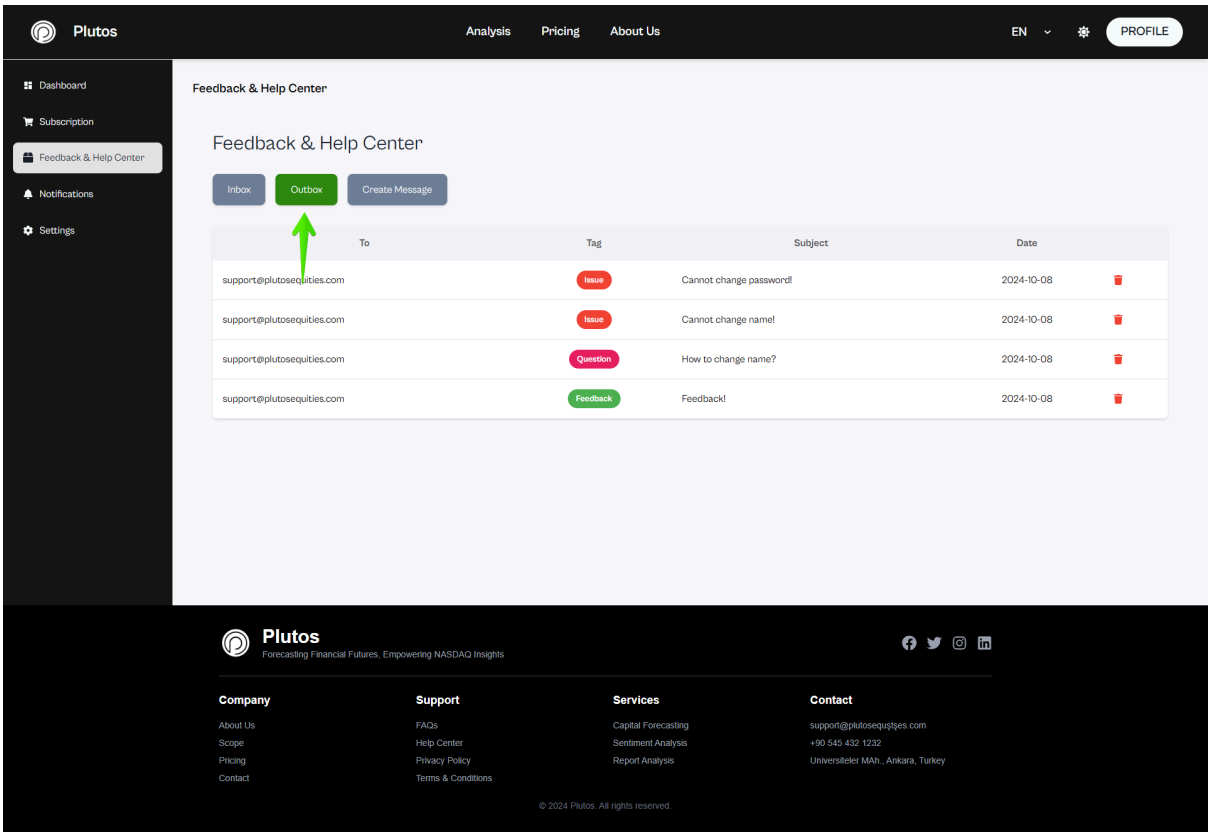
1. Here you see that your subscription change is confirmed. You can see your subscription by clicking on the Go to Subscription button shown by green arrow or pressing Subscription located in the sidebar.



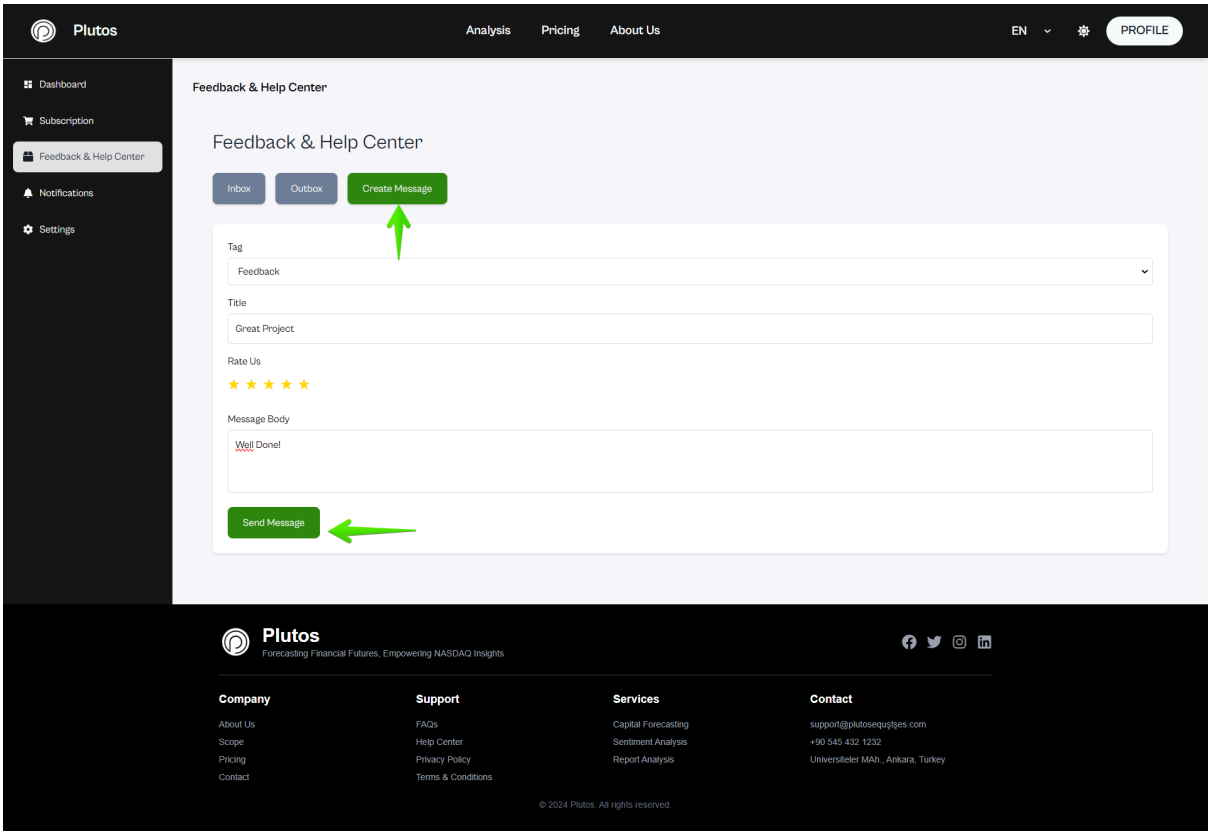
Third section of the profile page is Feedback & Help Center (press 1). This section is for the interaction between user and the system. It lets users give feedback, report issues or ask questions.

1. By default Inbox is displayed but you can also come to Feedback & Help Center Inbox anytime in this section by pressing Inbox button. (press 2)
2. In this inbox you only see the replies regarding your feedback, issues or questions coming from support@plutosequities.com.
3. If you want to delete, you can press the corresponding trash icon displayed at the end of each row.

Note: The notification(message) view logic will be discussed in details in the next section Notification.

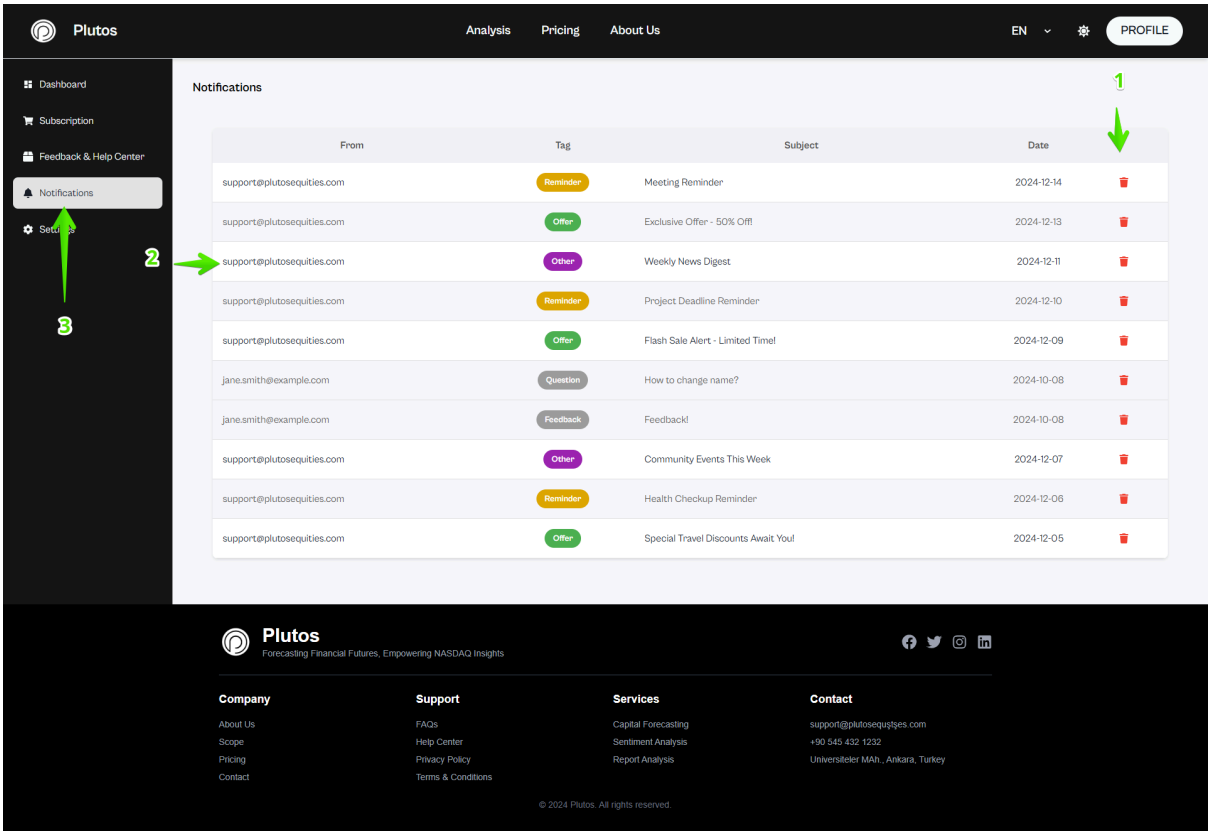


1. If you want to see the messages that you have sent/created to the plutos-support team, you can click on the Outbox button shown by the green arrow.



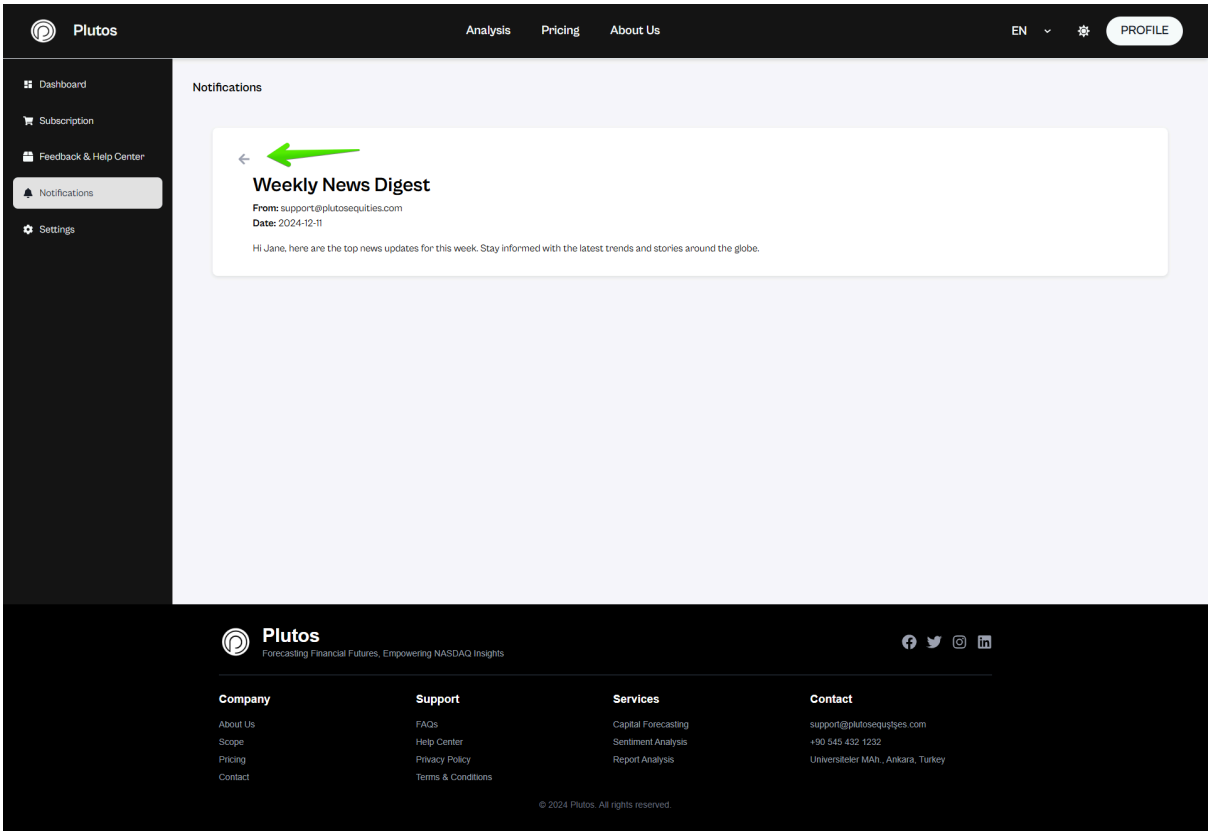
By pressing the Create Message button, you will see the frame in which you can create a message for plutos.(see green arrow at top)

1. This message can be a feedback, in this case a star rating bar displayed, an issue or a question.
2. After filling out the required fields. Press send message to send it to support. (see green arrow at bottom)

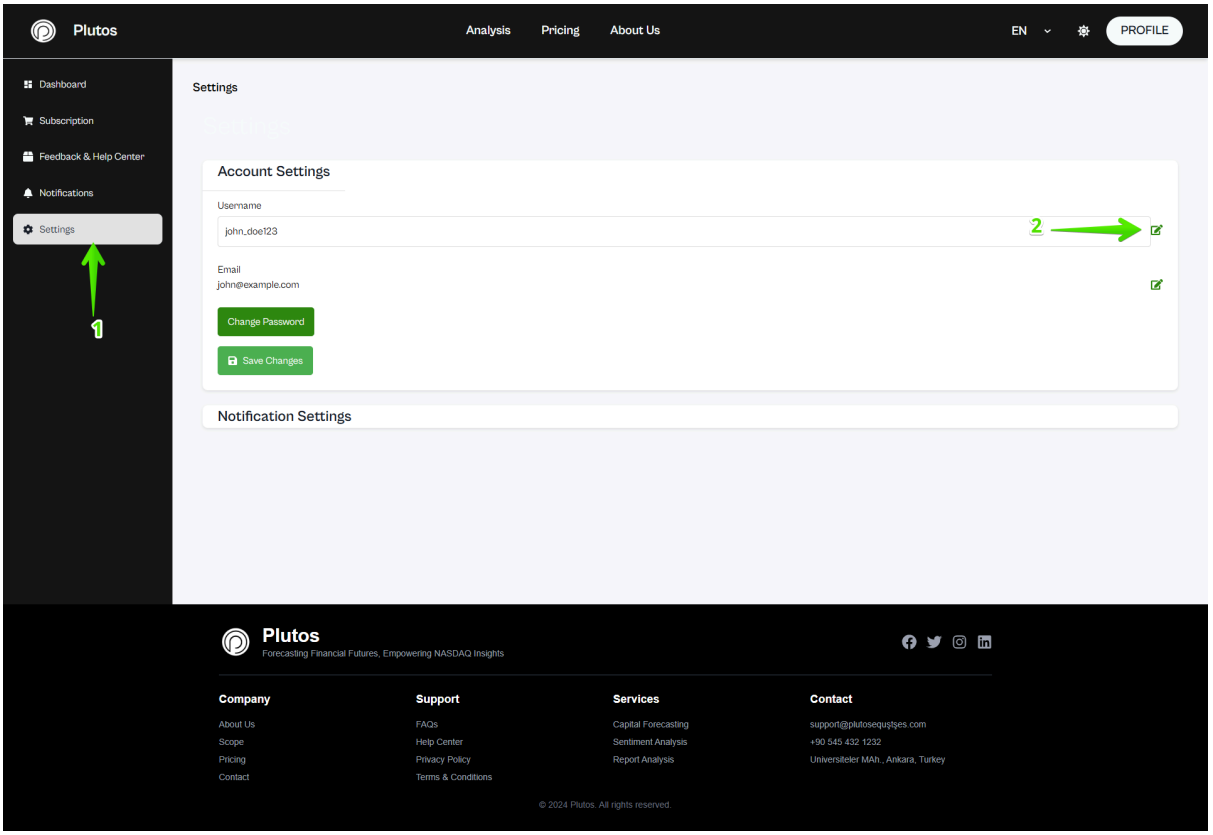


The fourth section of the profile page is Notifications, you can open this section by pressing Notification label on the sidebar (press 3). Here you see notifications, active (unread) ones are whiter than inactive ones (already read ones).

1. You can delete any notification by pressing the trash icon. (press 1)
2. To view a notification, simply click on its corresponding row. (press 2)



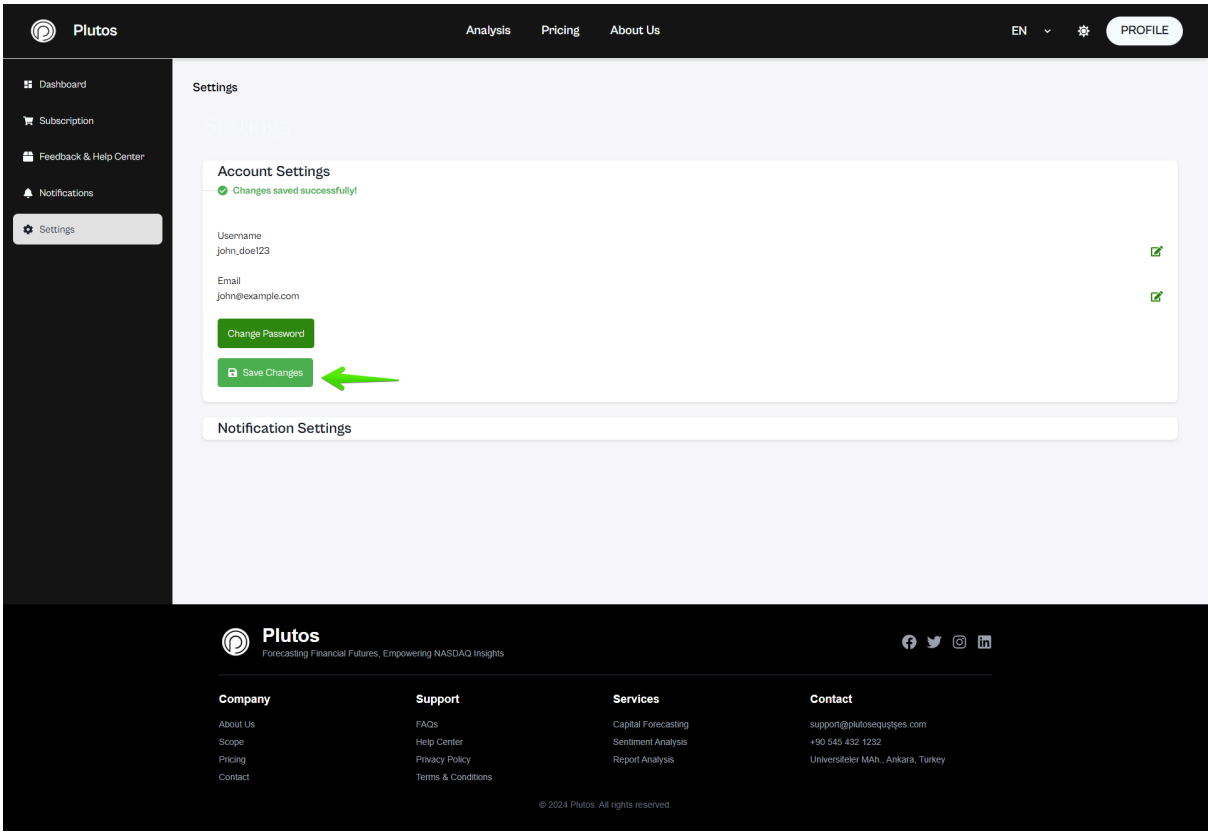
1. After pressing on a notification, you are able to see its content. To go back, click on Back buton shown by the green arrow.



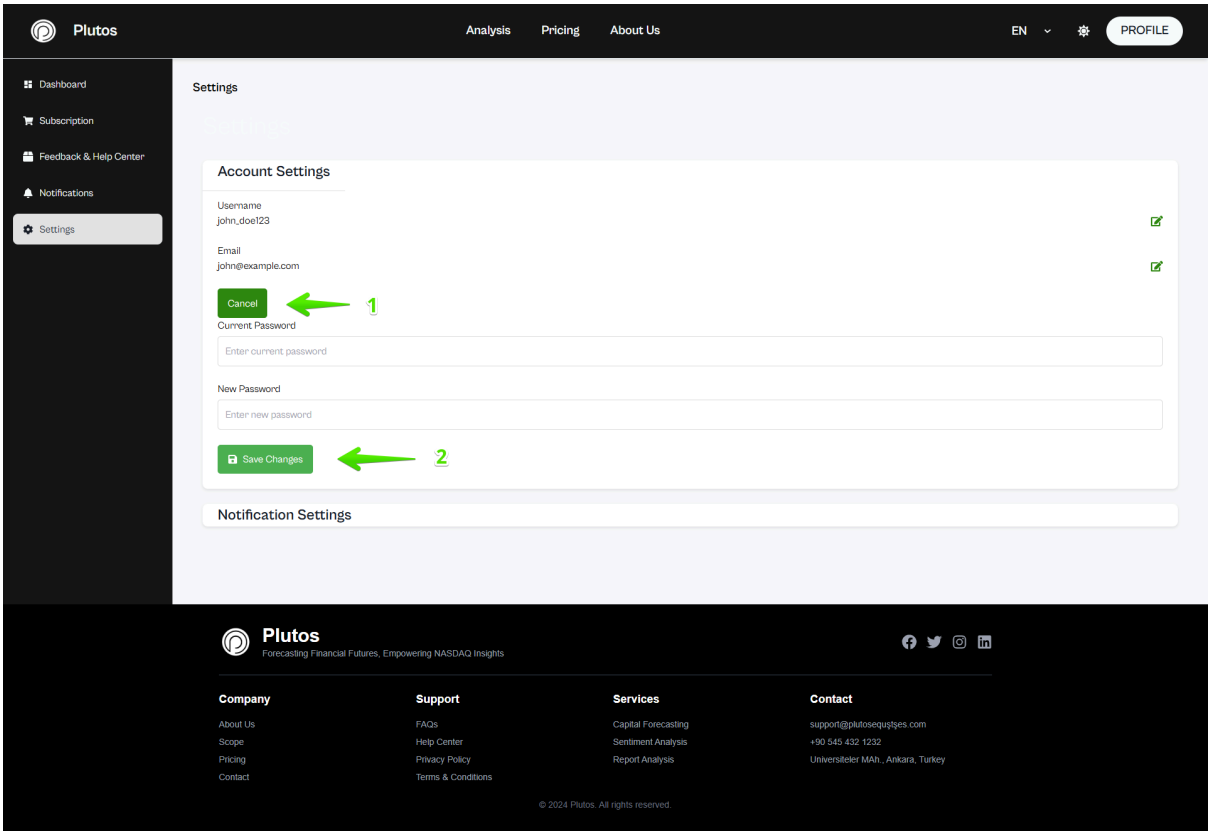
The last section of the profile page is Settings.

1. This section can be opened by pressing on its label in the sidebar. (press 1)

2. Press on Account Settings to see its content. It is a dropdown frame.
3. If you want to edit your name or email press the edit icon located in the right. (press 2)

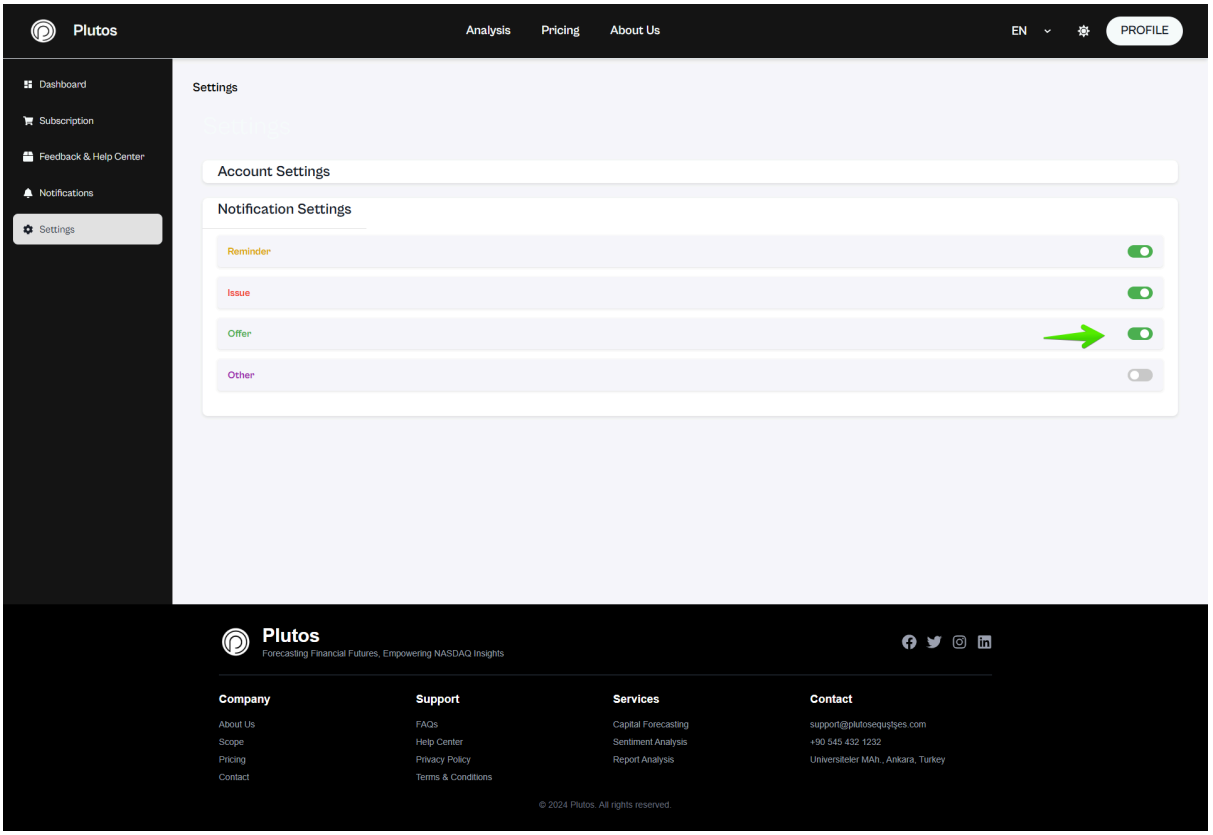


1. After the editing Press Save Changes button shown by the green arrow to save the edited versions. You will see a success message if everything is alright. If an error occurs, you will see an error message instead of a success message.
2. If you want to change your password, press the Change Password button.



Enter your current password and your new password. The current password must be correct in order to proceed.

1. You can abandon password changes by pressing the Cancel button. (press 1)
2. You can save password changes by pressing the Save Changes button. (press 2)



To see notification settings, press on Notification Settings label, it is a dropdown frame. If you want to receive email notifications turn on notifications by making it green (gray is for turned off notifications) (see green arrow). In addition, you may select specific types of notification (e.g. only turning off the other typed notification).

4. Other Analysis Elements

4.1. Consideration of Various Factors in Engineering Design

4.1.1. Constraints

4.1.1.1. Implementation Constraints

- **Diverse Data Sources:** The platform relies on integrating both structured data and unstructured data. Processing both structured and unstructured data is a challenging task.
- **Real-time Updates:** Implementing real-time data processing systems requires advanced technologies and architectures, such as streaming data pipelines, which can be complex to develop and maintain.
- **Unstructured Data Processing:** Extracting meaningful insights from unstructured data necessitates natural language processing techniques to obtain information and handle potential misinformation.
- **Data Accuracy and Reliability:** Ensuring the data collected is accurate and reliable is crucial for developing a good model.
- **Infrastructure Requirements:** Setting up and maintaining the necessary infrastructure (servers, databases, cloud services) adds layers of complexity to the implementation.
- **Model Maintenance:** Building a robust machine learning model is only the first

step. Continuous monitoring, retraining, and fine-tuning are required to maintain accuracy and adapt to evolving data patterns.

- **Robust Testing Methods:** Developing and implementing robust validation techniques, such as cross-validation, backtesting, and stress testing, is necessary to assess the model's performance under various scenarios.
- **Accuracy vs. Interpretability:** Balancing the need for high accuracy with model interpretability during the validation process is a significant constraint.
- **Data Leakage:** While developing models for financial data, it is important to prevent data leakage. It is especially hard to develop a time series model.
- **Time Constraints:** For evaluation of the model, it is important that the model development is completed before at least one quarterly report is made to measure the performance of the model in real life.

4.1.1.2. Economic Constraints

- **API Costs:** The platform uses APIs to access real-time financial data, historical stock prices, news sentiment, and market trends. Licensing and subscription costs for APIs, such as Alpha Vantage, Yahoo Finance, and similar services, are a significant expense. These costs increase with higher data usage as the project scales.
- **AWS Hosting Costs**: Hosting the platform on AWS involves expenses for essential services such as servers, storage, and data transfer. As the platform handles large volumes of data and runs ML models, the demand for computational power and storage capacity is expected to grow. This leads to higher expenses for essential cloud services, including those required for computation, data storage, and managing serverless operations.
- **Scalability Costs**: As the user base grows, the system must scale to handle more requests and higher data volumes. This includes upgrading server capacity storage limits and implementing load balancing, all of which increase AWS costs.
- **Data Storage Costs**: Maintaining historical financial data, market indices, and news articles requires significant storage resources. These storage requirements will grow over time, particularly as unstructured data like social media sentiment and earnings call transcripts are added to the dataset.

4.1.1.3. Ethical Constraints

- **Transparency of Predictions**: Ensuring the platform communicates how predictions are generated is a main ethical consideration. Users must have told the data sources and limitations involved in the forecasting process to avoid misinterpretation or over-reliance on the predictions.
- **Data Privacy and Security**: The platform handles potentially sensitive financial data, requiring strict data privacy measures. No user data or proprietary information obtained through API integrations or data partnerships will be shared with third parties without explicit consent.
- **Advisory Nature of Predictions**: Predictions generated by the platform are inherently probabilistic and subject to change based on changing market conditions. To prevent over-reliance, the platform will clearly communicate the limitations of the forecasts, ensuring users understand the data is advisory rather than definitive.
- **Prevention of Misuse**: Measures will be taken to ensure the predictions are used responsibly and within legal boundaries. Clear user agreements will outline acceptable

use policies, discouraging any activities that could lead to unfair advantages or violate ethical standards in financial decision-making.

4.1.1.4. Social Constraints

In the development of Plutos Equities, we have not identified any specific social constraints apart from two considerations. The first is ensuring the protection of any proprietary data used within the platform. The second is implementing safeguards to communicate the expected uncertainty in predictions, emphasizing that forecasts may not always be accurate and should not be the only basis for critical financial decisions.

4.1.2. Standards

- **Market Data Standards:** Usage of globally recognized data formats for market indices and stock prices, such as those provided by financial data providers like Bloomberg or Reuters.
- **General Data Protection Regulations**: Compliance is necessary for handling people. Ensuring data protection and privacy is important.
- **Cloud Infrastructure Standards:** AWS hosting should follow Well-Architected Framework principles, ensuring reliability, security, performance, and cost-efficiency.
- **Legal Standards**: Avoid using insider or illegal data, and follow basic financial data-sharing rules.
- **Focus on User Experience**: Design a simple, clear, and interactive interface (test it with actual users).
- **Clean Code Standards:** Writing clean code is crucial for scaling and reusing the code.
- **Testing Standards:** Developers and testers will write thoroughly unit and integration tests while continuing development. Tests are crucial.
- **Documentation Standards:** Documentation for the written development and user experience will be written for ease of use.

4.2. Risks and Alternatives

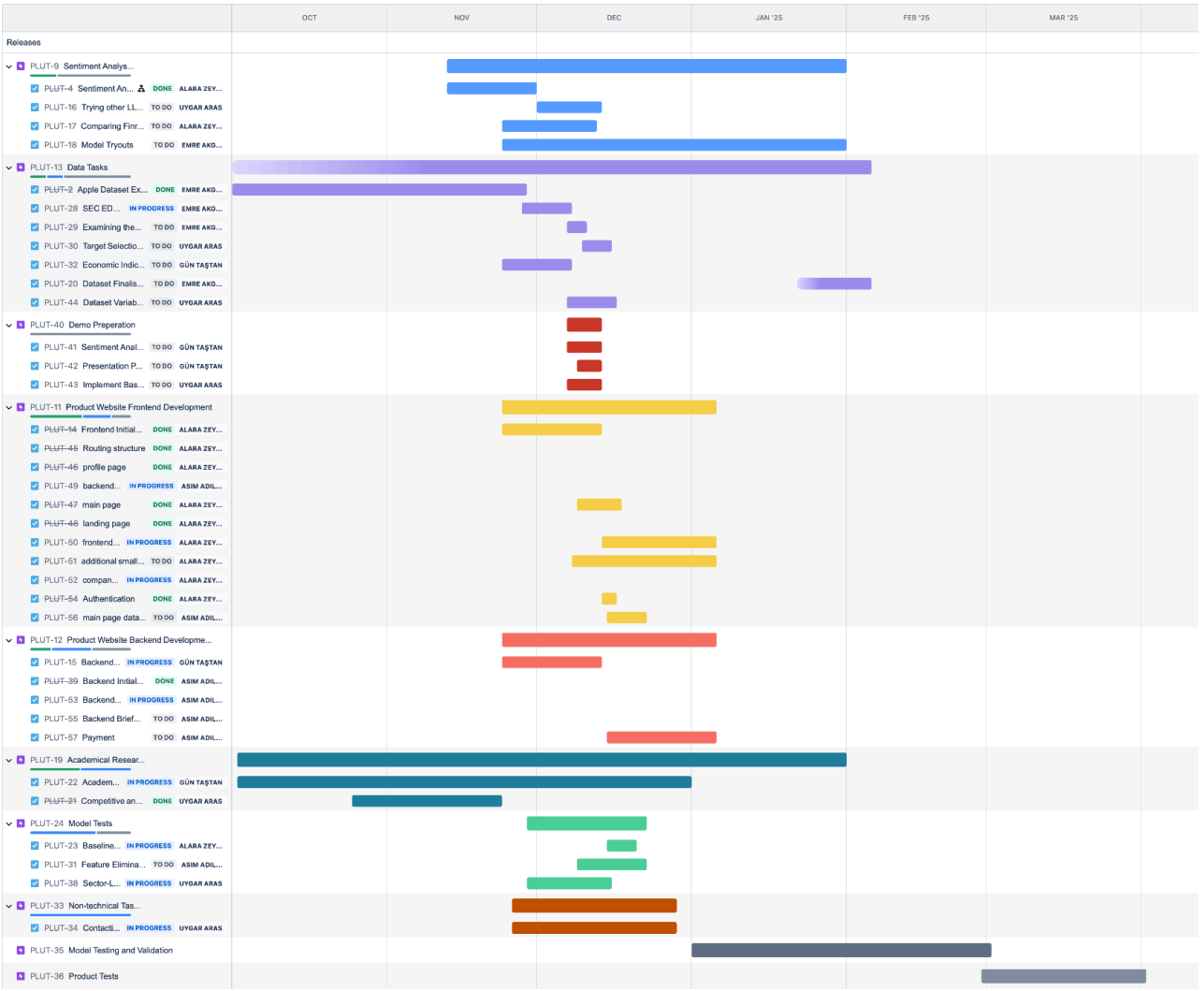
Plutos Equities relies on multiple data sources and machine learning models for capital forecasting. There is a risk that API data retrieval might fail or be incomplete. In such cases, alternative data sources and fallback mechanisms will be implemented.

Since the machine learning models are trained on limited historical financial data, the predictive accuracy may not meet initial expectations. In such scenarios, ensemble modeling techniques and manual adjustments will improve predictions.

The cloud infrastructure may face scalability challenges as the platform grows. In this scenario, we will implement modular architecture and adaptive scaling strategies.

Risk Name	Likelihood	Effect on the project	B Plan Summary
API Data Retrieval Failure	Medium	Platform unable to fetch financial data	Implement multiple data source fallbacks
Machine Learning Model Inaccuracy	Medium	Predictions may not meet accuracy requirements	Use ensemble modeling and manual calibration
Cloud Infrastructure Scalability	Medium	Performance bottlenecks and system limitations	Develop modular, adaptive cloud architecture
Sentiment Analysis Limitations	Low	Incomplete market sentiment interpretation	Enhance NLP models with additional training data
User Authentication Security	Low	Potential unauthorized access	Implement multi-factor authentication

4.3. Project Plan

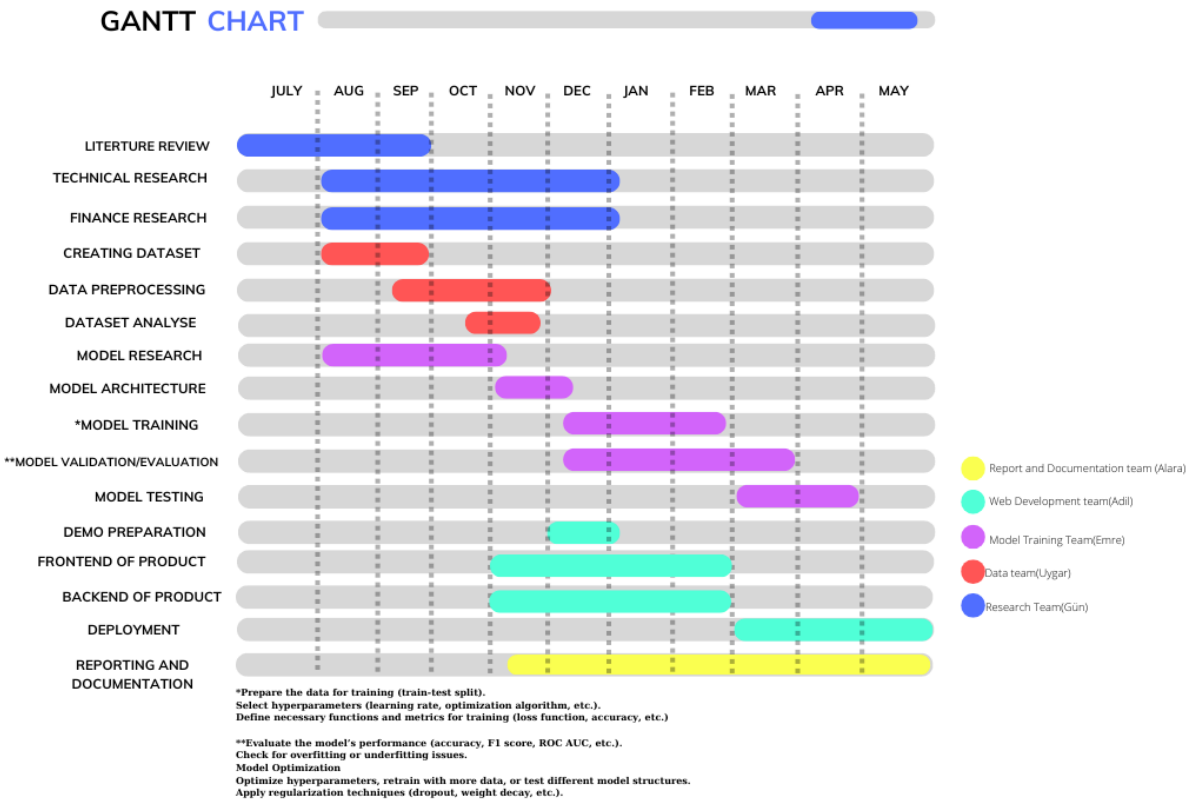


This Gantt chart which is a reflection of our JIRA Board outlines the detailed breakdown of tasks and their timelines, providing a clear overview of the project's progress so far and upcoming phases. The project started in **October** with initial tasks focused on **data analysis and sentiment analysis**. These tasks included comparing financial models, testing inputs, and preparing datasets, which were largely carried out through November and continued into early December. Concurrently, **dataset-related tasks** such as target selection, economic index evaluation, and dataset finalization have also been progressing, with most of them nearing completion as we move further into December.

Currently, the project is transitioning into its next phase, where efforts are shifting toward **demo preparation** and refining outputs for evaluation. **Frontend website development** has already started, covering structural design, implementation of the main page, and organizing components, which will continue through the upcoming months. Simultaneously, the **backend development phase** is beginning to take shape, with server and API setups planned to start soon and continue into early 2024.

As we move into **January and February**, the focus will turn to **model testing, validation, and integration**. These tasks will ensure that the models deliver accurate and reliable forecasts.

Meanwhile, **academic research** and **competitive analysis** will be completed in parallel to strengthen the theoretical and practical basis of the project. From **March to May**, the final phases of the project will include rigorous **model testing, validation, and deployment** to ensure a fully functional product. The process will conclude with **documentation and reporting**, finalizing all outputs for presentation and evaluation.



This Gantt chart that we have prepared at the beginning of our project and which we still follow provides a high-level view of the project timeline, broken into major phases spanning from **July to May**. The earlier stages of the project, between **July and October**, focused on foundational tasks such as the **literature review, technical research, and financial research**. These phases laid the groundwork for understanding market trends, identifying datasets, and determining methodologies for the project.

From **September through December**, significant progress was made on **data-related activities**, including dataset creation, preprocessing, and analysis. By the beginning of **December**, attention shifted to **model research and architecture**, where algorithms and frameworks are being explored and developed to ensure robust performance. As we move further into **December**, model training and validation are becoming the focus, with iterative improvements expected to continue into **February**.

Looking ahead, **January to April** will involve **frontend and backend development** of the product. The team will build the user interface, integrate backend systems, and ensure all

components align seamlessly with the forecasting models. During this time, **model testing** will also be prioritized to optimize outputs and address any issues.

From **March to May**, the final phases of the project will emphasize **model validation, testing, and deployment**, ensuring the product is reliable and ready for presentation. This will be accompanied by the preparation of final **documentation and reporting**, summarizing the results and delivering the completed product.

4.4. Ensuring Proper Teamwork

- All team members should join weekly meetings on time and actively take part in discussions to stay updated and aligned with the project goals.
- Each team member should actively contribute to all stages of the project, from planning to execution, and take responsibility for their tasks while helping with group decisions.
- Team members need to research their assigned topics carefully and share their findings. They should work together, help each other when needed, and ask the supervisor for guidance if problems arise.
- Tasks should be shared fairly, keeping in mind each member's skills, interests, and strengths, so everyone stays motivated and the work gets done efficiently.
- Team members should communicate openly and honestly to solve conflicts, give helpful feedback, and work smoothly as a group.
- Every team member is responsible for the quality of their work. The team should review all work together to make sure it meets high standards and is finished on time.
- Each member should keep a detailed logbook to record their tasks, progress, and challenges. The logbook should be updated regularly and reviewed by the supervisor to ensure accountability and track the team's development.

4.5. Ethics and Professional Responsibilities

In the development of Plutos Equities, professional and ethical issues that may arise are considered. Financial forecasting involves handling sensitive data, and maintaining the highest ethical standards is important to ensure trust and reliability. One of the primary concerns is data privacy. While Plutos Equities processes large amounts of financial data, including historical financial reports, market indices, and real-time news, no personal or proprietary user information will be collected or stored. The predictions generated by the platform will primarily rely on publicly available data, supplemented by proprietary data where necessary, ensuring compliance with data privacy regulations such as GDPR and maintaining transparency in data usage [6]. Measures will also be taken to secure data pipelines and prevent unauthorized access to the platform's systems.

Since the forecasts provided by Plutos Equities may influence investment and financial decisions, ensuring accuracy and transparency is crucial. The methodologies used for prediction will be openly communicated to users through detailed documentation, allowing them to understand the underlying processes. Additionally, disclaimers will clarify that the predictions are probabilistic in nature and should not be interpreted as guarantees. To address ethical concerns, safeguards will be implemented to prevent the misuse of predictions, such as for market manipulation or spreading misinformation. The platform will include user agreements specifying acceptable usage of the predictions and prohibiting unethical activities. Monitoring systems may be introduced to detect and flag any suspicious activities tied to the use of the forecast.

4.6. Planning for New Knowledge and Learning Strategies

For the project, we are developing a capital forecasting platform for the top 100 NASDAQ-listed companies. While we have already started gaining knowledge in key areas, including time-series forecasting, sentiment analysis, and structured data integration, we recognize that there is still significant room to expand and deepen our expertise as we continue working on the project.

We have already studied the fundamentals of time-series forecasting through Facebook Prophet documentation and tutorials, and some group members have begun implementing initial forecasting models for key financial metrics such as revenue and EPS. To build on this foundation, we will further explore advanced techniques such as Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), and Transformer architectures [7]. These models will be used to enhance the accuracy and scalability of our forecasting pipeline.

For sentiment analysis, we have initiated work with pre-trained transformer-based models like RoBERTa and BERT to analyze financial news and earnings call transcripts [8]. We are in the process of fine-tuning these models to improve their performance in identifying sentiment trends that can complement structured financial data. To continue advancing in this area, we plan to experiment with various natural language processing (NLP) techniques and optimize the integration of sentiment outputs with time-series data.

Our implementation efforts currently leverage Jupyter notebooks for collaborative development. Team members are now familiar with these tools, and we are actively using GitHub for version control and task tracking. To further enhance our collaborative workflow, we will use GitHub Projects to better manage assignments and monitor our progress. We have also started incorporating feedback from early prototype testing to refine our development practices.

In terms of data integration, we have begun retrieving structured financial data using APIs like Yahoo Finance and Alpha Vantage. This has allowed us to experiment with data-cleaning techniques and feature engineering processes using tools like pandas and NumPy [9]. Moving forward, we will focus on automating data pipelines to handle large datasets in real time while ensuring data consistency and accuracy.

We are also in the process of researching financial reporting standards and the expectations of our target users, including investors, auditors, and analysts. This involves reviewing industry reports, studying academic research, and gathering insights from stakeholders. As we continue this research, we aim to ensure that our platform aligns with user needs and provides actionable insights.

Although we have made significant progress, we acknowledge the need to continue our learning journey. Specifically, we plan to deepen our understanding of advanced machine learning techniques, experiment with new data integration methods, and refine our approach to model evaluation and testing. By regularly consulting academic advisors and industry experts, we aim to ensure that our efforts remain aligned with best practices and emerging trends in financial forecasting [10].

Through ongoing learning, collaborative development, and iterative testing, we are committed to delivering a reliable and user-friendly platform by the project deadline. This

process will ensure that our work not only meets the expectations of our stakeholders but also sets a strong foundation for future enhancements and innovations.

5. Glossary

1. AI-Driven Analytics: The use of artificial intelligence to analyze data, detect patterns, and make predictions or decisions, particularly in forecasting financial metrics.
2. Capital Expenditures (CapEx): Funds used by a company to acquire, upgrade, and maintain physical assets such as property, industrial buildings, or equipment. Relevant to understanding a company's financial health in forecasting models.
3. Capital Forecasting: The process of estimating a company's future financial performance, including metrics such as revenue, operating costs, and earnings per share (EPS).
4. Cash Flow Analysis: The examination of the inflows and outflows of cash in a business is critical for understanding liquidity and operational efficiency.
5. Compound Annual Growth Rate (CAGR): A measure used to express the mean annual growth rate of an investment over a specified period of time.
6. Earnings Before Interest, Taxes, Depreciation, and Amortization (EBITDA): A measure of a company's overall financial performance, used as an alternative to net income in some cases.
7. Earnings Per Share (EPS): A key financial metric calculated as a company's profit divided by the number of outstanding shares, indicating the profitability of a company on a per-share basis.
8. Ensemble Methods: Machine learning techniques that combine multiple models to improve predictive performance. These methods are applied to financial forecasting to achieve more accurate results.
9. Equity Analysis: The process of evaluating a company's stock performance, financial health, and future prospects, often forming the basis of investment decisions.
10. Feature Engineering: The process of selecting and transforming raw data into useful features that enhance the performance of machine learning models used in financial analysis.
11. Financial Ratios: Metrics derived from financial statements that are used to evaluate a company's performance, such as debt-to-equity ratio and return on equity (ROE).
12. LSTM (Long Short-Term Memory): A type of recurrent neural network (RNN) architecture designed to process and predict data sequences over time, particularly useful for time-series forecasting in finance.
13. Market Capitalization (Market Cap): The total value of a company's outstanding shares, calculated by multiplying the share price by the number of shares. It serves as a measure of company size and value.
14. Market Sentiment: The overall attitude of investors and market participants toward a particular security or financial market, often gauged using sentiment analysis.
15. Natural Language Processing (NLP): A branch of AI that focuses on enabling computers to understand and process human language. In this project, it is used to analyze text data from financial news and reports.
16. Operating Expenses (OpEx): The costs of running a business's core operations, excluding direct costs of goods sold (COGS).
17. Price-to-Earnings Ratio (P/E Ratio): A valuation metric that compares a company's share price to its earnings per share, often used to assess whether a stock is over- or under-valued.
18. Prophet Model: An open-source time-series forecasting tool developed by Facebook, often used for trend analysis and seasonal data, and applied in capital forecasting in this project.
19. Quarterly Reports: Financial documents companies release every three months detailing their performance. Forecasting these reports is a primary goal of the project.

20. Return on Investment (ROI): A measure of the profitability of an investment, calculated as the net profit divided by the initial investment cost.
21. Sentiment Analysis: Using NLP to determine the emotional tone of textual data. In this project, it is used to gauge market sentiment from news and social media.
22. Stock Market Forecasting: The process of predicting the future performance of stocks or indices, often using machine learning and statistical models.
23. Time-Series Forecasting: A method of predicting future values based on historical data points. Models like LSTM and Prophet are used for this project.
24. Volatility: A statistical measure of the dispersion of returns for a given security or market index, often used to assess investment risk.
25. SEC Edgar: The Electronic Data Gathering, Analysis, and Retrieval system of U.S. Securities and Exchange Commission.

6. References

[1] M. Jensen and W. Wu, "Machine learning in financial forecasting: A systematic review," *Journal of Financial Analytics*, vol. 5, no. 2, pp. 35–47, 2022.

[2] Y. Zhang, L. Chen, and K. Zhou, "The impact of unstructured data on predictive accuracy in financial systems," *IEEE Transactions on Computational Finance*, vol. 3, no. 4, pp. 123–135, Dec. 2021.

[3] G. Smith and A. K. Lee, "Real-time financial forecasting: Combining historical and dynamic inputs," *Applied Finance Research Journal*, vol. 14, no. 3, pp. 70–85, 2023.

[4] S. Rahman and H. Gupta, "Explainable AI for finance: Addressing the transparency gap in machine learning predictions," *IEEE Access*, vol. 10, pp. 12045–12059, 2022.

[5] C. Nguyen, J. Kim, and R. Patel, "Real-time data integration in financial models: Challenges and approaches," in *Proceedings of the IEEE Conference on Data Science*, Singapore, 2023, pp. 567–574.

[6] D. Smith and R. Brown, "Ethical considerations in financial data handling and privacy," *Journal of Financial Data Ethics*, vol. 8, no. 3, pp. 45–59, 2022.

[7] H. Zhao, Z. Zhang, and Y. Liu, "Enhancing time-series forecasting with LSTM and Transformer models: A comparative study," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5608–5619, Dec. 2021.

[8] Y. Liu, X. Li, and W. Yang, "Sentiment analysis using BERT and RoBERTa for financial text mining," in *Proceedings of the IEEE International Conference on Artificial Intelligence*, Singapore, 2022, pp. 340–345.

[9] A. Gupta and M. Rahman, "Data integration and preprocessing techniques for financial forecasting," in *Proceedings of the International Conference on Data Science and Applications*, New York, NY, 2023, pp. 129–135

[10] L. Zhang, "Best practices in model evaluation and testing for financial forecasting," *IEEE Access*, vol. 10, pp. 23100–23110, 2022.