



نمو

Nomu





نموا | Nomu

IT 496: Graduation Project Report
Product Release-1

Prepared by

Student Name:	Student ID:
Norah Aljedai	443200841
Asma Alshilash	443200439
Alhanouf Aldakel Allah	443200999
Doaa Aldobai	443203882
Raghd Ahmed Hassan	443204743

Supervised by
Dr. Qatrunnada Alsmail
Dr. Yousra Saud Almathami

Table of Contents:

.1	INTRODUCTION
 9
2.	THE PROBLEM
	<i>Main Product Objectives</i> 10
4	BACKGROUND
 15
4.4	DOMAIN KNOWLEDGE
 15
4.4.1	<i>investment</i> 15
4.4.2	<i>Stock</i> 16
4.4.3	<i>stock market</i>
4.4.4	<i>Investment Portfolio</i>
4.4.5	<i>Application Program Interface (API)</i>
4.4.6	<i>Simulation System</i>
4.4.7	<i>Machine learning</i>
4.4.8	<i>Gamification</i> 18
4.4.9	<i>Technical analysis</i>
4.4.10	<i>Moving Averages</i> 18
4.4.11	<i>Autocorrelation (ACF) & Partial Autocorrelation (PACF)</i> 21
4.4.12	<i>Autoregression (AR)</i> 22
4.5	TERMINOLOGY
 23
5	LITERATURE REVIEW
 25
5.4	STOCK MARKET SIMULATION 25
5.5	GAMIFICATION 26
5.6	COMPETITIVE PRODUCT ANALYSIS 28
5.7	SUMMARY OF LITERATURE REVIEW: 34
6	SYSTEM DESIGN AND DEVELOPMENT
 36
6.4	METHODOLOGY
 36
6.5	SYSTEM REQUIREMENTS
 38
6.5.1	<i>System Users</i> 38
6.5.2	<i>Requirements Elicitation and Analysis</i>
•	<i>Interviews</i> 38
•	<i>Questionnaires</i>
•	<i>Conclusion</i>
6.5.3	<i>User Interactions</i>
6.5.4	<i>Roadmap and Product Backlog</i>
•	<i>Product Backlog</i>
6.6	SYSTEM DESIGN
 56
6.6.1	<i>Architectural Diagram</i> 56
6.7	CLASS DIAGRAM /DFD 57
6.8	COMPONENT LEVEL DESIGN
 59
6.8.1	<i>Access Library of Tutorials</i> 60
6.8.2	<i>Interactive Daily Challenge</i>
6.8.3	<i>Video Lesson Completion</i>
6.8.4	<i>AI-Based Stock Prediction</i>
6.8.5	<i>Virtual Portfolio Performance</i>
6.9	DATA DESIGN
 70
6.9.1	<i>Data Models</i>
6.9.2	<i>Data Collection and Preparation</i>
 72

Semester-2 1446H (Spring 2025)

6.10 INTERFACE DESIGN.....	75
6.10.1 Site Map of Nomu	75
6.10.2 Implemented UX Guidelines	78
6.11 GAMIFICATION AND REWARD MECHANICS.....	83
7 SYSTEM IMPLEMENTATION	88
7.4 KEY IMPLEMENTATION STEPS	88
7.5 KEY IMPLEMENTATION PROCEDURES.....	90
7.6 CHALLENGES	132
7.7 GITHUB REPOSITORY	134
8 SYSTEM EVALUATION.....	170
8.1 USER ACCEPTANCE TESTING	170
8.2 DEMOGRAPHICS OF PARTICIPANTS	170
8.3 QUESTIONNAIRE RESULTS	170
8.3.1 System Usability Scale (SUS) Results.....	171
8.3.2 Nomu Objectives Results	173
8.4 QUALITY ATTRIBUTES (NFR TESTING)	175
8.5 DISCUSSION.....	177
9 SYSTEM TESTING.....	179
9.4 USER ACCEPTANCE TESTING	179
9.4.1 Testing Design and Methodology.....	179
9.4.2 Demographics of Participants.....	180
9.4.3 Questionnaire/Interview Results	183
9.5 DISCUSSION.....	186
10 CONCLUSIONS AND FUTURE WORK	189
10.4 CONCLUSIONS	189
10.5 GLOBAL AND LOCAL IMPACT.....	189
10.6 PROBLEMS AND CHALLENGES ENCOUNTERED DURING SOFTWARE DEVELOPMENT	190
10.7 LIMITATIONS OF THE SYSTEM.....	191
10.8 MAIN CONTRIBUTIONS OF THE NOMU.....	192
10.9 FUTURE WORK	193
11 ACKNOWLEDGEMENTS	194
12 REFERENCES.....	196
13 APPENDICES.....	200
13.4 APPENDIX A: INTERVIEWS	200
13.5 APPENDIX B: QUESTIONNAIRE	204
13.6 APPENDIX C: UAT QUESTIONNAIRE	211
13.7 APPENDIX D: UAT QUESTIONNAIRE	215
13.8 APPENDIX E: OBJECTIVES EVALUATION QUESTIONS AND RESULTS	220
13.9 APPENDIX F: GITHUB REPOSITORY AND JIRA PROJECT	225

List of Tables:

Table 1: Competitive Product Analysis	33
Table 2:Scrum Team.....	37
Table 3:Product Backlog.....	45
Table 4: Access Library of Tutorials	60
Table 5: Interactive Daily Challenge	62

Semester-2 1446H (Spring 2025)

Table 6: Demographics of Participants	170
Table 7: SUS Questions and Results	172
Table 8: Participant's SUS scores and their interpretation	172
Table 9: Interpretation of SUS score	173
Table 10: NFR Testing Results	176
Table 11: First Interview Outline	200
Table 12: Second Interview Outline	201
Table 13: Third Interview Outline	202
Table 14: Fourth Interview Outline	203
Table 15: First Interview Outline	215
Table 16: Second Interview Outline	216
Table 17: Third Interview Outline	217
Table 18: Forth Interview Outline	218
Table 19: Fifth Interview Outline	219
Table 20: objectives evaluation Questions and Results	220

List of Figures:

Figure 1: stock market.....	16
Figure 2: Charting a 50-Day Simple Moving Average.....	19
Figure 3: MACD with Histogram.....	20
Figure 4: Trade the MACD	21
Figure 5:Autoregressive models with increasing model orders, $p \in \{0,1,2,3,\}$. AR(0) is white noise, while AR($p > 0$) exhibit autocorrelation with increasingly complex dependencies between xt and previous observations.....	22
Figure 6: Tadawul Simulator.....	28
Figure 7: Investmate application	29
Figure 8: Trading Game Application	30
Figure 9: Stock Trainer Application	31
Figure 10: Stock Market Simulator Game application	32
Figure 11: Nomu use case diagram.....	43
Figure 12:Roadmap.....	44
Figure 13: Architectural Diagram.....	57
Figure 14: Class diagram	59
Figure 15: Flowchart of the Interactive Daily Challenge.....	63
Figure 16: Video Lesson Completion	65
Figure 17: Flowchart Virtual Portfolio Performance	69
Figure 18: Nomu ER diagram	70
Figure 19: Nomu tree structure.....	71
Figure 20:Nomu site map	75
Figure 21:Usability and Ease of Use UX Guideline	78
Figure 22: Clear navigation UX Guideline	79
Figure 23:Feedback and Confirmation UX Guideline	80
Figure 24: Consistency and Standards	81
Figure 25:Smart Validation and Security.....	82
Figure 26:Daily Challenge interfaces	83
Figure 27: Flashcard Interfaces	84

Semester-2 1446H (Spring 2025)

Figure 28: Second Trial reward screen	85
Figure 29: Video lessons interfaces	86
Figure 30 Firebase Dependencies	Error! Bookmark not defined.
Figure 31 Initializing Firebase.....	Error! Bookmark not defined.
Figure 32 Sign-Up page	Error! Bookmark not defined.
Figure 33 Sign-Up page 2	Error! Bookmark not defined.
Figure 34 Login page 1	Error! Bookmark not defined.
Figure 35 Login page 2	Error! Bookmark not defined.
Figure 36 Login page 3	Error! Bookmark not defined.
Figure 37 forgot_password_page	Error! Bookmark not defined.
Figure 38 UserProfilePage	Error! Bookmark not defined.
Figure 39 edit_profile_page	Error! Bookmark not defined.
Figure 40 Tutorial Section 1.....	Error! Bookmark not defined.
Figure 41 Tutorial Section 2.....	Error! Bookmark not defined.
Figure 42 Flash card 1.....	Error! Bookmark not defined.
Figure 43 Flash card 2.....	Error! Bookmark not defined.
Figure 44 Daily question 1	Error! Bookmark not defined.
Figure 45 Daily question 2	Error! Bookmark not defined.
Figure 46 Daily question 3	Error! Bookmark not defined.
Figure 47: coin manager.....	Error! Bookmark not defined.
Figure 48: hasEarnedCoinsForLesson	Error! Bookmark not defined.
Figure 49: checkAndRewardCoins() for lessons	Error! Bookmark not defined.
Figure 50: checkAndRewardCoins() for tutorials	Error! Bookmark not defined.
Figure 51 :hasEarnedCoinsForLesson().....	Error! Bookmark not defined.
Figure 52: Daily Challenge Rewards	Error! Bookmark not defined.
Figure 53: fetchEducationFavorites()	Error! Bookmark not defined.
Figure 54: Removing Favorites	Error! Bookmark not defined.
Figure 55: Toggling Lesson Favorites	Error! Bookmark not defined.
Figure 56: toggleFavorite()	Error! Bookmark not defined.
Figure 57: FavoriteVideoIconState.....	Error! Bookmark not defined.
Figure 58: toggleFavorite()	Error! Bookmark not defined.
Figure 59: Dependencies & Configuration	Error! Bookmark not defined.
Figure 60: load the API key	Error! Bookmark not defined.
Figure 61: GeminiService	Error! Bookmark not defined.
Figure 62 Age distribution	181
Figure 63:Gender distribution	181
Figure 64:Education Level	182
Figure 65:Experience with Investment Applications.....	182
Figure 67 Learning Tools (Flashcards & Daily challenges)	184
Figure 68 Ease Learning	184
Figure 69 Recognizability of Icons.....	185
Figure 70 Willingness to Use the System	185
Figure 71: Pie chart of question 1 (age distribution)	204
Figure 72: Pie chart of question 2 (gender distribution)	204
Figure 73: Pie chart of Question 3 (Education Level distribution)	205
Figure 74: Pie chart of Question 4 (Current Investment Knowledge distribution)	205
Figure 75: Pie chart of Question 5 (Interest in Learning Investment distribution)	206

Semester-2 1446H (Spring 2025)

Figure 76: Bar chart of Question 6 (Preferred Investment Type distribution).....	206
Figure 77: Bar Chart of Question 7 (Motivation for Learning Investment distribution)	207
Figure 78: Bar Chart of Question 8 (Challenges Faced in Learning Investment distribution)	207
Figure 79: Pie Chart of Question 9 (Encouragement to Start Investing via Learning & Simulation)	208
Figure 80: Pie Chart of Question 10 (Preferred Method for Learning Investment distribution)	208
Figure 81: Pie Chart of Question 11 (Preference for Learning via Website or Application)	209
Figure 82: Pie Chart of Question 12 (Preference for Platform Language distribution).....	209
Figure 83: Pie Chart of Question 13 (Impact of Game-like Experience on Learning Continuation)	210
Figure 84: Age distribution	211
Figure 85: Gender distribution	211
Figure 86: Education Levels.....	212
Figure 87: Prior Experience with Finance Applications	212
Figure 88: Ease of Navigation.....	212
Figure 89: Error Message Clarity	213
Figure 90: Flashcards & Daily Questions Impact	213
Figure 91: Feature Understanding	213
Figure 92: Icon and Label Recognition.....	214
Figure 93: Willingness to Reuse the System.....	214



01 Introduction



1. Introduction

Investing plays a fundamental role in financial growth and long-term economic stability. However, many individuals especially beginners find the investment world difficult to access due to limited financial literacy, misconceptions about needing large amounts of capital, and fear of financial loss. These challenges discourage many from taking their first step, preventing them from benefiting from potential financial opportunities.

In alignment with Saudi Vision 2030, which emphasizes strengthening financial literacy and empowering individuals to participate actively in the economy [1], the Nomu application is proposed as a solution to address this gap. Nomu provides a safe and risk-free learning environment that enables beginners to explore basic investment concepts, practice using historical market data, and build confidence through interactive and engaging educational content.

By simplifying financial knowledge and offering hands-on practice, Nomu supports users in making informed decisions and understanding how financial markets operate. This initiative not only reinforces national efforts to enhance financial awareness but also contributes to global movements aimed at expanding financial inclusion and accessible education.

2. The Problem

Many individuals find the world of investing overwhelming and difficult to access, especially when they lack the financial knowledge required to get started. Misconceptions about needing large amounts of capital, the fear of losing money, and the complexity of financial terminology create a major barrier for beginners. As a result, many people avoid investing altogether, missing opportunities for personal financial growth.

A concrete example of this challenge can be seen in current investment platforms such as Tamra [2], which require users to answer detailed questions about financial goals, risk tolerance, and investment experience before they can begin. For beginners with no prior knowledge, these questions can feel intimidating and confusing, discouraging them from taking even the first step toward investing.

This problem is even more pronounced in Saudi Arabia, where only 30% of adults are considered financially literate [3]. With a population exceeding 35 million, many individuals lack the confidence and foundational knowledge needed to engage in the stock market. Additionally, the limited availability of Arabic-language investment tools further restricts access to educational resources, leaving many potential investors unprepared to navigate global financial platforms.

The specific part of this problem that our project addresses is the lack of a safe, simple, and accessible way for beginners to learn and practice investing without financial risk. Many existing platforms focus on actual trading rather than preparation, leaving a gap for users who need foundational knowledge and hands-on experience before entering real markets.

Our proposed application, Nomu, directly targets this gap by offering a risk-free simulation environment designed for beginners. Through gamified lessons, interactive tutorials, and AI-powered simulations using historical data, Nomu enables users to explore investment concepts, try strategies, and build the confidence needed to make informed financial decisions.

3. Objectives

Nomu is being developed to address the gap between financial knowledge and practical investment experience among beginners. Many individuals are interested in learning how to invest but lack a safe, simplified environment that allows them to practice without financial risk. The application provides an interactive educational platform designed to help users understand foundational concepts and build confidence before entering real financial markets.

The application aims to solve several key challenges faced by new investors, such as limited financial awareness, difficulty understanding investment principles, and fear of loss due to the absence of a safe practice environment. Nomu also addresses the lack of accessible Arabic content that simplifies investment topics and the need for a practical tool that enables hands-on trading experience without using real money.

A wide range of users will benefit from Nomu, including beginner investors who require practical training before investing, students and young adults seeking to enhance their financial literacy, and individuals with limited budgets who wish to explore investment strategies without exposure to risk. Moreover, the application is suitable for anyone who wants to understand how financial markets work and learn how to build and manage an investment portfolio in a simplified and practical way.

Main Product Objectives

- Realistic Stock Market Simulator
 - Uses historical market data to simulate buying, selling, and tracking the performance of a virtual portfolio.
- Interactive Learning Lessons
 - Includes structured lessons, videos, and flashcards to simplify investment concepts.
- Gamified Learning Experience
 - Features daily challenges, a coin reward system, and progress dashboards to enhance motivation and engagement.
- Virtual Portfolio Management
 - Helps users track their portfolio performance, understand risk management, and reset their portfolio to test new strategies.
- Favorite Stocks, Lessons & Flashcards
 - Allows users to save preferred lessons, flashcards, and stocks for quick access, enabling a more personalized and efficient learning experience.

4. Scope

Nomu is an Android mobile application designed to educate beginner investors and individuals with limited budgets in Saudi Arabia through a risk-free, engaging, and interactive learning platform. It features a virtual stock market simulator that uses historical market data, enabling users to practice investment strategies, view virtual portfolios, reset portfolio, and gain hands-on experience in portfolio management without financial risks.

The application is designed to cater to Arabic-speaking users, ensuring accessibility and relevance to the target audience while promoting an engaging and effective learning experience. To enhance user

Semester-2 1446H (Spring 2025)

engagement, Nomu incorporates gamified features such as coins, and leaderboards, encouraging consistent learning and participation. Additional tools like account management, a personalized dashboard for monitoring progress, and options to save preferred topics or stocks further enrich the user experience. Furthermore, the application provides notifications to keep users motivated and informed about new updates, ensuring they stay engaged. A support center is included to address troubleshooting and ensure seamless usage.

Built with the Flutter framework and Dart programming language, Nomu offers a responsive and compatible interface optimized for Android devices. The application is solely education, excluding real-time financial transactions, live market data, algorithmic trading, or advanced analytics, making it an ideal platform for beginner investors. By blending education, gamification, and practical simulation, Nomu empowers users to confidently explore the fundamentals of investing in a practical and enjoyable way.

5. Product vision statement

For beginners and hesitant investors with limited budgets

Who lack the knowledge and confidence to invest

The Nomu is a financial education and simulation mobile application

That helps users build confidence in making investment decisions by simulating real-world market scenarios using historical data

Unlike other available investment applications like Tamra [2] **Our** product focuses on providing a risk-free, educational environment to help users understand investment concepts and practice decision-making without real financial risk.

6. Development Methodology

The development of Nomu followed an Agile software development methodology, which emphasizes iterative progress and continuous integration of feedback. Using the Scrum framework, the project was divided into short development sprints that enabled the team to gradually implement features, refine them, and ensure that the application evolved in alignment with user needs. The process began with analyzing user requirements through surveys and market research, which guided the identification of both functional and non-functional requirements. Tools such as Jira [4] were used for task management and tracking progress, while GitHub [5] supported version control and collaborative development.

From a technical perspective, the application's interface was developed using Flutter and Dart [6], while Firebase [7] was utilized for authentication, secure data storage, and portfolio management. Visual assets were designed using Canva [8], and the educational content was curated from trusted and reputable financial sources to ensure accuracy and reliability. These materials were adapted into custom-designed instructional videos created from scratch, utilizing PlayHT [9] to generate the AI voiceovers for the audio narration.

For the stock market simulation component, historical stock data was collected from the Tadawul platform [10]. The dataset underwent extensive cleaning and preprocessing to remove missing or inaccurate values and prepare it for analytical use. Additionally, the team developed a prediction model capable of analyzing price trends to forecast whether a stock's value is likely to rise or fall the next day. This model was integrated into the simulation to provide users with contextual, training-oriented recommendations to improve investment decision-making.

Finally, the application underwent several rounds of performance testing, usability testing, and model evaluation to ensure system stability, prediction accuracy, and smooth user experience. This iterative and structured methodology enabled the development of a practical, educational, and engaging solution tailored specifically for beginner investors.

7. contributions

The main contribution of Nomu lies in providing a unified, accessible, and interactive educational platform that bridges the gap between financial knowledge and real-world investment experience for beginners. Instead of exposing users to financial risks or overwhelming them with complex market tools, Nomu offers a safe environment where individuals can learn, practice, and build confidence through a realistic stock market simulator, interactive lessons, and gamified challenges. This holistic approach directly addresses the lack of financial literacy, fear of investing, and limited availability of Arabic educational resources in the field.

The solution helps users overcome the initial barriers to investing by simplifying financial concepts and enabling hands-on practice with historical stock data. Nomu empowers individuals, especially young adults, students, and people with limited budgets to make informed decisions and understand risk management without using real money. By enhancing users' financial awareness and decision-making skills, the application contributes to building a more financially informed community. On a broader level, the solution supports Saudi Arabia's Vision 2030 goals of strengthening economic participation and promoting financial literacy, while also presenting a scalable model that can positively impact global communities seeking accessible financial education tools.

Nomu differentiates itself from existing market solutions by combining Arabic-learning content, risk-free stock simulation, and gamified engagement within a single mobile application tailored to beginners. Unlike typical stock simulators or financial platforms that target experienced investors or rely on live markets, Nomu focuses exclusively on education, simplification, and practical training. Its integration of interactive lessons, daily challenges, and personalized dashboards creates a learning experience that has not been offered in this form before, making the solution both novel and highly relevant to underserved users in the region.

8. Report organization

This report provides a comprehensive overview of the development of the Nomu project. Chapter 1 introduces the research problem, project objectives, and the motivation behind the application. Chapter 2 establishes the theoretical background and domain context, while Chapter 3 critically reviews related literature and existing solutions in the market.

The technical development is detailed in the subsequent chapters: Chapter 4 outlines the system requirements and elicitation process; Chapter 5 presents the system design, including architecture, data models, and user interface specifications; and Chapter 6 describes the system implementation, covering key technical procedures and challenges.

Chapter 7 evaluates the system through user acceptance testing and quality assessments. Finally, Chapter 8 concludes the report by summarizing the main contributions of Nomu and outlining potential directions for future work. The report closes with acknowledgements, references, and supporting appendices.



02 Background



4 Background

Investment is a cornerstone of personal financial growth and economic stability, yet it remains inaccessible for many beginners due to misconceptions about high capital requirements, complex financial concepts, and fear of financial loss. The problem is further exacerbated by low financial literacy rates and a lack of accessible educational resources tailored to specific cultural and linguistic needs, such as those of Arabic-speaking users. Without practical tools and guidance, many individuals hesitate to explore investment opportunities, limiting their ability to build wealth and contribute to broader economic development.

To address these challenges, Nomu offers a comprehensive educational platform that combines simulation-based learning, gamification, and predictive stock insights. By providing a risk-free simulated trading environment powered by machine learning and historical market data, users can practice investment strategies, explore portfolio management, and gain confidence without financial risk. Gamified tutorials and interactive challengers make learning engaging, while the platform simplifies investment education, empowering users to make informed decisions and bridging the gap between theoretical knowledge and practical application.

4.4 Domain knowledge

4.4.1 investment

Investment involves allocating resources into assets like stocks, bonds, real estate, or businesses to generate profit and build wealth. Key types include:

- Stocks: Ownership in companies, high risk/reward.
- Bonds: Lower-risk fixed-income securities.
- Real estate: Rental income or long-term appreciation.
- Mutual funds: Diversified, professionally managed assets.
- Commodities: Gold, oil, etc., traded for profit.
- Cryptocurrencies: High-risk, high reward (e.g., Bitcoin).

In Saudi Arabia, investment is central to Vision 2030, which promotes economic diversification through sectors like tech, tourism, and renewable energy. Initiatives like NEOM and regulatory reforms aim to attract foreign and domestic investors.

However, financial literacy gaps persist globally. A 2019 MIT study found 60% lack investment knowledge, leading to poor financial decisions and reduced chances of independence [12].

Nomu addresses this by offering an educational platform with machine learning-powered market simulations. It simplifies financial concepts, enables risk-free strategy practice, and bridges the gap between theoretical knowledge and real-world application, empowering users to make informed decisions.

4.4.2 Stock

A stock represents ownership in a company, entitling the holder to a share of its profits and voting rights in key decisions. Traded on exchanges like the New York Stock Exchange (NYSE), the London Stock Exchange (LSE), or Tadawul in Saudi Arabia., its performance is tracked using six core data points, which are stored in the database:

1. **Date**: The trading day when data is recorded.
2. **Open**: The stock's starting price, reflecting initial market sentiment.
3. **High**: The day's peak price, indicating maximum demand.
4. **Low**: The day's lowest price, showing market pressure.
5. **Close**: The final traded price, critical for assessing daily performance.
6. **Volume**: Total shares traded, signaling investor activity and interest.

This data helps investors make informed decisions about buying or selling stocks based on the stock's market performance, price changes, and trading volume.

4.4.3 stock market

The stock market is a platform where publicly traded company shares are bought and sold, as shown in the flowchart of companies, brokers, a central exchange, and investors. By issuing shares through an Initial Public Offering (IPO), companies raise capital from the public, while brokers match buy and sell orders at the stock exchange; once matched, trades are executed, and ownership of shares passes to new holders. This mechanism underpins how businesses secure funds for expansion while offering investors returns via dividends or capital gains. Prices shift in response to supply and demand, influenced by factors like market sentiment, company performance, and economic conditions. Through mandated financial disclosures, the system promotes transparency and supports economic growth by channeling capital into productive ventures.

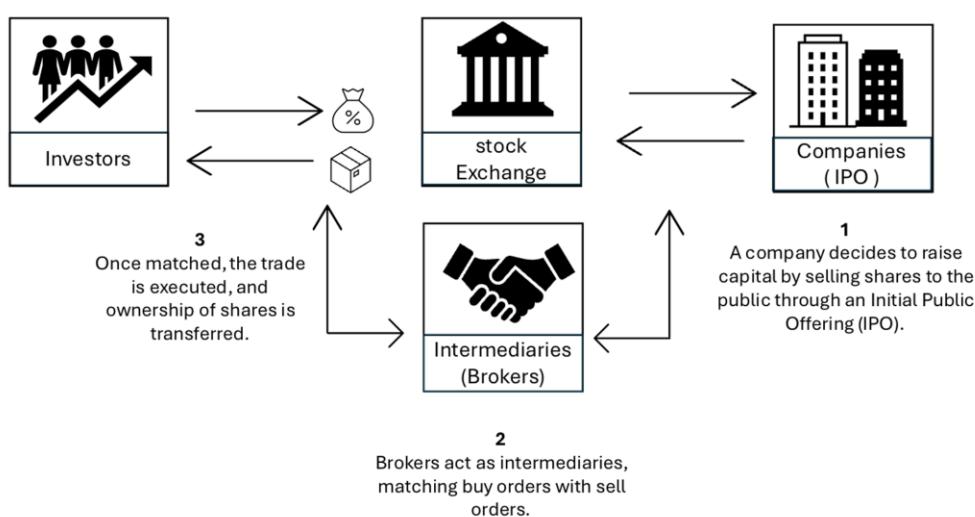


Figure 1: stock market

Nomu prioritizes stock market participation because of its foundational role in personal and economic well-being. Investing in stocks offers an accessible entry point for individuals of all experience levels, fostering crucial financial skills such as risk management, market analysis, and strategic decision-making. Over time, these skills enable investors to diversify portfolios, build wealth, and contribute to the broader economy—objectives that the stock market structure, as depicted in the flowchart, is designed to facilitate.

4.4.4 Investment Portfolio

The investment portfolio in the Nomu application provides a centralized overview of a user's financial holdings, designed to track progress toward investment goals and potential returns. It aims to achieve financial growth and meet specific investment objectives. The portfolio includes user stocks and cash, offering a comprehensive summary of investments and their current value.

In Nomu, the portfolio helps users balance risks and optimize performance. The application enables users to monitor investments continuously and adjust strategies based on market performance and portfolio changes.

This portfolio provides an overview of the user's financial status, allowing informed decisions regarding buying or selling stocks and working toward stable and sustainable returns over the long term.

4.4.5 Application Program Interface (API)

APIs enable applications to communicate, allowing data exchange without rebuilding systems. In the Nomu application, APIs integrate historical market data for realistic simulations, while machine learning (ML) enhances learning by analyzing trends and simulating market behavior.

Tadawul Reference Data API: Provides static and descriptive information about listed companies and financial instruments, such as symbols, sectors, ISIN codes, share details, and corporate actions.

Gemini API: Supports natural language interaction through an AI-powered chatbot that assists users in learning financial concepts and navigating the application.

ML further enriches the experience by predicting market movements and offering feedback, helping users practice informed decision-making in risk-free scenarios. This blend of data and ML creates an immersive learning environment for mastering financial strategies.

4.4.6 Simulation System

A simulation system mimics real-world processes or environments, enabling users to learn, experiment, or practice without real-world risks. Stock market simulations replicate trading operations virtually, letting users buy, sell, and manage stocks using historical data—with financial risk. These tools help users test strategies, understand market behavior, and build confidence before real trading.

Semester-2 1446H (Spring 2025)

Nomu leverages such simulations by offering a platform where users practice trading with real historical data. Machine learning (ML) predicts market trends, allowing users to test strategies, adapt to market changes, and refine decision-making skills in a risk-free environment. This approach bridges theoretical knowledge and practical investing, equipping users with the experience and confidence needed for real-world financial markets.

4.4.7 Machine learning

Nomu uses machine learning to analyze market data and predict stock trends through classification algorithms, categorizing price movements as "upward" or "downward" based on technical indicators like MACD, simplifying investment decisions by highlighting opportunities and risks.

The ARIMA (Autoregressive Integrated Moving Average) model forecasts short-term stock prices by analyzing historical trends and seasonality. It includes three components:

- p** (Autoregressive term): Past observations used for prediction
- d** (Differencing term): Number of differencing to achieve stationarity
- q** (Moving Average term): Past forecast errors included in the model

Autoregression (AR), a part of ARIMA, uses past values to predict future prices. By combining ARIMA with classification algorithms, Nomu provides numerical forecasts and directional signals, enabling investors to simulate strategies and make data-driven decisions.

4.4.8 Gamification

Gamification uses game elements like points, rewards, and challenges to enhance engagement in non-game contexts. Research by Hamari, Koivisto, & Sarsa (2014) indicates that gamification generally has positive effects, though its success depends on context and user characteristics [19]. On platforms like Nomu, it supports investment learning by allowing users to practice trading in a risk-free environment using virtual currency. Users engage with interactive simulations, receive feedback on their decisions, and track their progress in refining investment skills. This approach transforms learning into an interactive experience, simplifying complex financial topics and building confidence through immersive, hands-on practice. Studies, such as Bitrián, Buil, & Catalán (2021), have shown that gamified features enhance motivation, increase perceived ease of use, and improve user attitudes toward financial management applications, reinforcing gamification's effectiveness in financial education [18].

4.4.9 Technical analysis

Technical analysis is a method of evaluating statistical trends in trading activity, typically involving price movement and volume. It is used to identify trading and investment opportunities.

4.4.10 Moving Averages

In finance, a moving average (MA) is a stock indicator commonly used in technical analysis. The reason for calculating the moving average of a stock is to help smooth out the price data by creating a constantly updated average price.

Semester-2 1446H (Spring 2025)

• Mathematical Formula:

• Simple Moving Average (SMA):

is a calculation that takes the arithmetic mean of a given set of prices over a specific number of days in the past. See Figure 3: Charting a 50-Day Simple Moving Average for an example of how the Simple Moving Average is plotted on a stock price chart.

$$SMA = \frac{P_1 + P_2 + P_3 + \dots + P_n}{n}$$



Figure 2: Charting a 50-Day Simple Moving Average.

Image by Sabrina Jiang © Investopedia 2021

Where P_1, P_2, \dots, P_n are the closing prices for a number of periods (e.g., 10 days), and n is the number of periods.

• Exponential Moving Average (EMA):

is a weighted average that gives greater importance to the price of a stock in more recent days, making it an indicator that is more responsive to new information.

$$EMA_{Today} = (Value_{Today} \times \left(\frac{Smoothing}{1+Days} \right)) + EMA_{Yesterday} \times \left(1 - \left(\frac{Smoothing}{1+Days} \right) \right)$$

In "Nomu," moving averages (SMA and EMA) are used to help users identify trends in the market more clearly. Users can use these averages in the simulation to discover upward or downward trends based on market movements.

4.4.10.1 Moving average convergence/divergence (MACD)

Moving average convergence/divergence (MACD) is a technical indicator to help investors identify price trends, measure trend momentum, and identify entry points for buying or selling. Moving average convergence/divergence (MACD) is a trend-following momentum indicator that shows the relationship between two exponential moving averages (EMAs) of a security's price.

- **Mathematical Formula:**

$$MACD = 12 - \text{Period EMA} - 26 - \text{Period EMA}$$

- **MACD Signals**

The MACD line is calculated by subtracting the 26-period EMA from the 12-period EMA. The calculation creates the MACD line. A nine-day EMA of the MACD line is called the signal line, plotted on top of the MACD line, which can function as a trigger for buy or sell signals.

Traders may buy the security when the MACD line crosses above the signal line and seller shorts the security when the MACD line crosses below the signal line. MACD indicators can be interpreted in several ways, but the more common methods are crossovers, divergences, and rapid rises/falls.

- **Histogram**

MACD is often displayed with a histogram (see Figure 4 below) that graphs show the distance between MACD and its signal line. If MACD is above the signal line, the histogram will be above the MACD's baseline or zero line. If MACD is below its signal line, the histogram will be below the MACD's baseline. Traders use the MACD's histogram to identify peaks of bullish or bearish momentum, and to generate overbought/oversold trade signals.

The MACD histogram is a visual representation of the difference between the MACD and its nine-day EMA—not highs and lows. The **Error! Hyperlink reference not valid.** is positive when the MACD is above its nine-day EMA and negative when the MACD is below its nine-day EMA. The point on the histogram where momentum is zero is the zero line.



Figure 3: MACD with Histogram

Semester-2 1446H (Spring 2025)

If prices change rapidly, the histogram bars grow longer as the speed of the price movement—its momentum—accelerates and shrinks as price movement decelerates.

- **Crossover Strategy**

A crossover occurs when the signal and MACD line cross each other. The MACD generates a bullish signal when it moves above its own nine-day EMA and triggers a sell signal (bearish) when it moves below its nine-day EMA.



Figure 4: Trade the MACD

Nomu will use the MACD indicator to predict buy and sell signals, helping users make informed investment decisions. By analyzing MACD crossovers and momentum shifts, the platform will provide real-time insights, guiding users on optimal entry and exit points in their simulated trades.

4.4.11 Autocorrelation (ACF) & Partial Autocorrelation (PACF)

ACF (Autocorrelation Function) measures the correlation between current values and past values, helping identify cyclical patterns and trends in time series data.

PACF (Partial Autocorrelation Function) is similar to ACF but eliminates intermediate effects, showing the direct relationship between current values and specific past values.

In Nomu, ACF and PACF are used to identify predictable patterns in stock price movements. These insights help fine-tune the parameters (p and q) in the ARIMA model, improving stock price predictions for better investment decisions.

4.4.12 Autoregression (AR)

Autoregressive (AR) models predict future values based on a linear combination of past values, assuming that past behavior influences future outcomes. The AR(p) model is defined as:

$$x_t = \alpha + \sum_{i=1}^p \beta_i x_{t-i} + \varepsilon_t$$

Where $\beta = \{\beta_1, \dots, \beta_p\}$ are the parameters, α is a constant bias, and ε_t is white noise. An AR(0) process is simply random noise with no dependencies between time points. As p increases, the model becomes more complex, incorporating more past observations to predict future values. See Figure 6 for synthetic data from AR processes of increasing order.

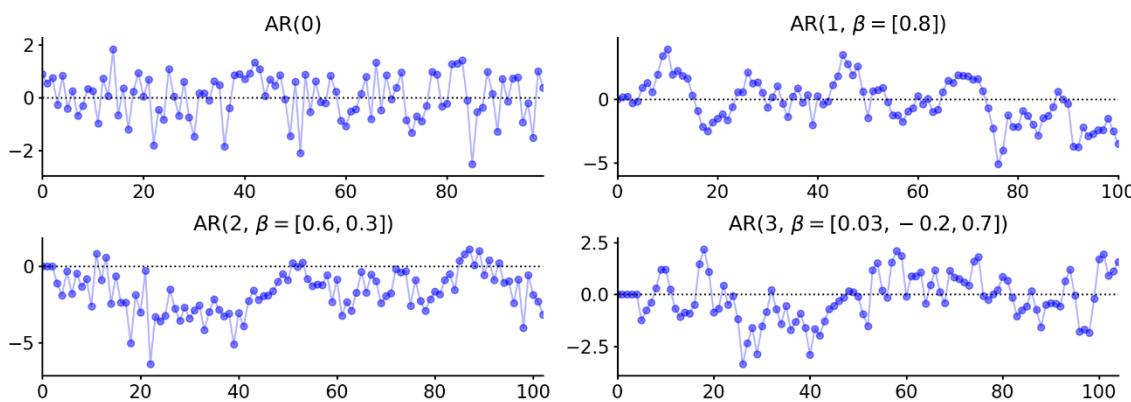


Figure 5: Autoregressive models with increasing model orders, $p \in \{0, 1, 2, 3\}$. AR(0) is white noise, while AR($p > 0$) exhibit autocorrelation with increasingly complex dependencies between x_t and previous observations..

Nomu leverages AR models to forecast stock prices by analyzing historical trends. By applying AR models, Nomu can generate more accurate predictions, helping users simulate investment strategies and make informed decisions based on the behavior of stocks over time.

4.5 Terminology

I. Bearish

- **Definition:** A market condition where asset prices are declining. Investors expecting a decline are "bearish."

II. Bullish

- **Definition:** A market condition where asset prices are rising. Investors expecting a rise are "bullish."

III. Trend

- **Definition:** The general direction of a time series over time. It can be:
 - **Upward:** Prices rise.
 - **Downward:** Prices fall.
 - **Stationary:** No long-term movement

IV. Seasonality

- **Definition:** Repeating patterns in a time series at regular intervals, often influenced by external factors like time of year, seen in markets, sales, and weather.

V. Zero line

- **Definition:** A reference level in technical indicators like MACD, used to determine market conditions (bullish or bearish) based on its position above or below.



03 Literature Review



5 Literature Review

This chapter explores previous research relevant to our project, focusing on stock market simulations and gamification in financial education. By reviewing key studies, we aim to identify essential requirements, justify our approach, and position our project within the broader academic landscape. We will analyze the methodologies used in these studies, highlighting their similarities and differences compared to our project. This analysis will help uncover research gaps and opportunities for further improvement, providing a strong theoretical foundation for our work.

5.4 Stock Market Simulation

Stock market simulations play a crucial role in investment education, allowing users to develop financial literacy and decision-making skills in a risk-free environment. Several studies have explored the impact of these simulations on learning outcomes and skill development.

Jankowski & Shank (2010) conducted a comparative analysis of online stock trading simulators, examining both free and paid platforms. Their study evaluated 16 websites that enable users to create investment portfolios and track their performance, concluding that while free simulators offer robust features, paid versions provide additional technical support and advanced functionalities. The study emphasizes the importance of selecting a simulator that aligns with educational objectives, balancing cost and feature availability [13].

Building on this, Dressler et al. (2016) investigated the impact of stock market simulation games on student engagement and skill development. Their case study revealed that simulations significantly enhance critical thinking, decision-making, and financial market understanding. By promoting collaborative learning, these tools allow students to apply theoretical concepts in practical scenarios, reinforcing financial education through experiential learning. The study underscores the value of integrating stock market simulations into academic curricula with well-structured instructional design [17].

Further advancing this discussion, Patil (2024) explored the role of stock market simulators in developing financial analytical skills within technical education. Over a 60-day experiment involving 50 students, the study observed substantial improvements in financial analysis and investment decision-making. While the research confirms that simulations enhance financial literacy and practical decision-making, it also identifies a gap in long-term studies, suggesting the need to assess the sustainability of these benefits over extended periods [18].

5.5 Gamification

Gamification has emerged as a powerful tool for enhancing user engagement and motivation across various domains, including financial education. Researchers have examined its effectiveness in different contexts, emphasizing its role in improving learning experiences and behavioral outcomes.

Hamari, Koivisto, & Sarsa (2014) conducted a comprehensive literature review on gamification, analyzing empirical studies from multiple fields. Their findings indicate that while gamification generally has positive effects, its success depends on contextual factors and user characteristics. The study highlights the necessity of carefully designing gamified systems to maximize their impact in diverse environments [19].

Expanding on this, Caponetto, Earp, & Ott (2014) reviewed 120 research papers published between 2011 and 2014, tracing the evolution of gamification in education. Their study revealed that the conceptual clarity of gamification has improved over time, distinguishing it from Game-Based Learning (GBL), where digital games are used explicitly for teaching. The findings suggest that gamification has gained recognition as an independent educational strategy, reinforcing its value in academic settings [17].

In the financial sector, Bitrián, Buil, & Catalán (2021) examined the impact of gamification on personal financial management (PFM) applications. Analyzing data from 208 Mint app users, they applied partial least squares structural equation modeling to evaluate how gamification satisfies users' psychological needs for competence and autonomy. Their results show that gamified features enhance motivation, increase perceived ease of use and usefulness, and improve user attitudes toward financial management applications. The study suggests that fintech companies and banks can leverage gamification to boost user engagement and adoption of financial tools [18].

A more targeted study by Pal, Indapurkar, & Gupta (2021) explored the moderating role of gamification in shaping financial behavior. Using structural equation modeling (SEM) with AMOS-21, they analyzed data from individuals who use financial applications for saving and investment. Their findings indicate that gamified elements significantly influence financial planning activity (FPA) and financial attitude (FA), enhancing user engagement in financial decision-making. However, the study found that gamification does not significantly impact the relationship between financial self-efficacy (FSE) and financial management behavior (FMB), suggesting that self-efficacy in financial management operates independently of gamified interventions [19].

Building on the user experience aspect, Baptista and Oliveira (2017) investigated the unified theory of acceptance and use of technology in the context of gamified mobile banking and investment apps. Their research highlighted that gamification acts as a critical antecedent to user intention, significantly increasing the "fun" factor which, in turn, positively influences the perceived ease of use. This is particularly relevant for novice investors, as it suggests that gamifying the onboarding process can reduce the intimidation often associated with complex financial interfaces [20].

Furthermore, in the specific context of trading simulations, Looyestyn et al. (2017) emphasized the importance of "safe failure" environments. Their systematic review suggests that gamified

Semester-2 1446H (Spring 2025)

simulations allow users to experiment with high-risk decisions without real-world financial consequences, thereby fostering a deeper understanding of market mechanics. This "sandbox" approach was found to be essential for bridging the gap between theoretical knowledge and practical application, as it permits users to learn from mistakes in a controlled setting before committing real capital [21].

Finally, Rodrigues et al. (2016) explored the relationship between gamified social elements—such as leaderboards and peer comparisons—and long-term application loyalty. Their study found that social interaction features within financial apps not only drive initial adoption but are crucial for sustaining long-term user retention. By creating a community-driven environment where users can track their progress relative to peers, financial applications can maintain higher levels of engagement over extended periods, preventing the "drop-off" often seen in educational apps [22].

By synthesizing these studies, we gain valuable insights into the role of stock market simulations and gamification in financial education. The findings underscore the effectiveness of simulation-based learning in enhancing financial literacy and decision-making while highlighting gamification's ability to increase motivation and engagement. These insights will inform our project, helping us design an effective and engaging educational tool that aligns with proven strategies in financial learning.

5.6 Competitive Product Analysis

1- Tadawul Simulator



The Application is a trading simulation app that provides a virtual trading experience, replicating the actual Saudi Stock Market. It allows users to test trading strategies in a realistic environment with an easy-to-use interface and interactive charts that help monitor and analyze market performance. Users can also review their investment decisions, making it an effective educational tool for practicing real-world trading in the capital market.

The app offers a comprehensive simulation of trading all listed securities in the Saudi Stock Exchange, with a quick and easy registration process. It provides users with a fixed virtual capital of SAR 100,000, enabling them to trade without any real financial risk. Additionally, it features real market data with a 5-minute delay, supports multiple order types, and automatically calculates commission fees. Moreover, the app allows users to generate financial reports when needed, making it an ideal tool for those looking to learn trading and refine their investment skills in a safe and realistic environment. [23]

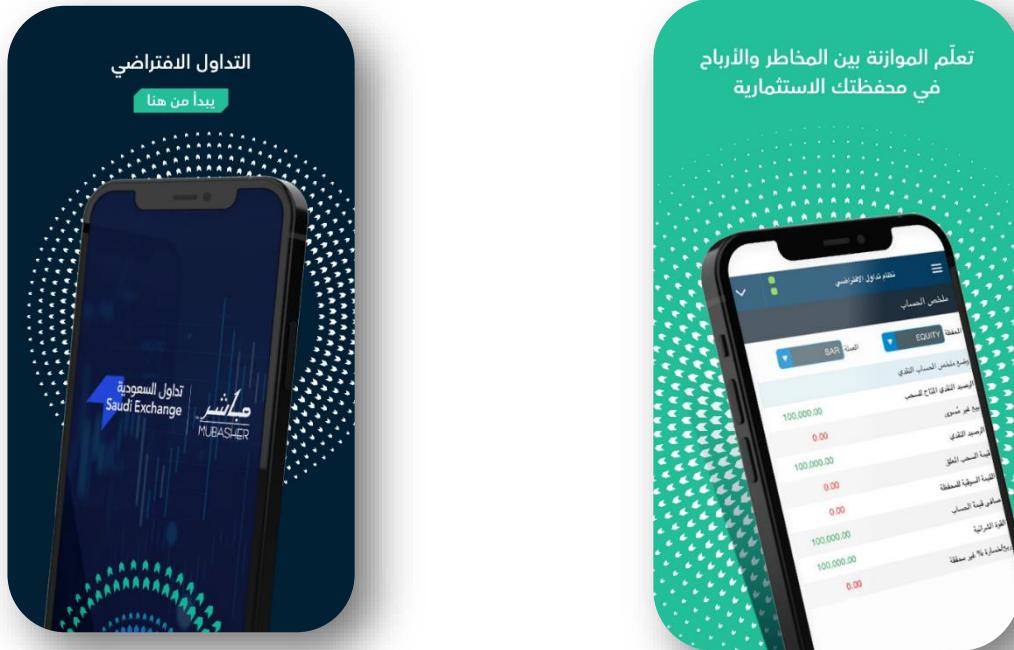


Figure 6: Tadawul Simulator

Semester-2 1446H (Spring 2025)

2- Investmate — Learn to trade



The Application is an educational app designed to teach users the basics of CFD (Contracts for Difference) trading. The app offers a comprehensive set of courses, trading strategy tips, quizzes, and a financial glossary. Whether you're a beginner or looking to deepen your knowledge of trading, Investmate provides the tools to help you learn at your own pace.

The app offers personalized learning through a feed that includes courses, interactive quizzes, educational videos, and a comprehensive glossary, helping you learn everything you need in an organized way. It allows you to complete lessons in just a few minutes and take interactive quizzes to check your progress, making learning both fun and effective. Thanks to its mobile-friendly design, you can learn anytime and anywhere that suits you.

Investmate also provides comprehensive content with simple and detailed explanations of complex financial concepts, covering various aspects of CFD trading, including stocks, commodities, and indices. Additionally, the app features a straightforward and clear interface with all content presented in cards, along with a pin feature that lets you return to saved materials at any time. [24]

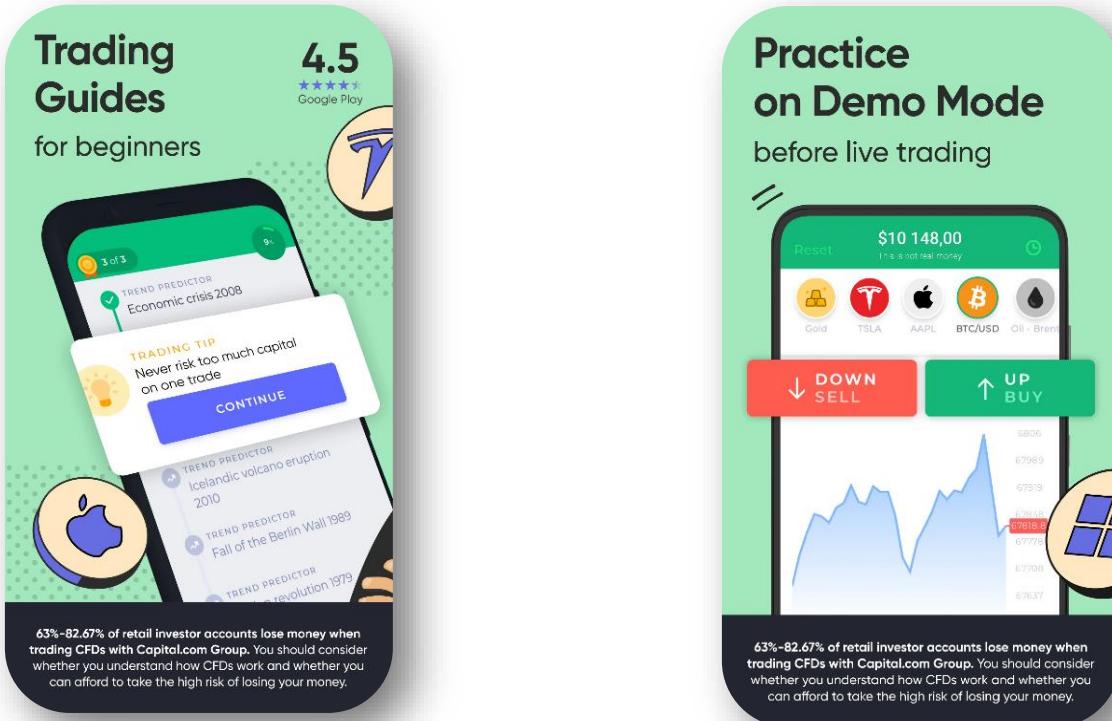


Figure 7: Investmate application

3- Trading Game: Stock Market Sim



The Application is an educational app designed to help users learn and practice stock trading and investing using live market data. The app allows users to join over 3 million students, offering a set of tools to learn and practice real trading without the need for a finance degree.

The key features of *Trading Game* include the Trading Academy, which offers over 90 lessons suitable for all levels, from beginners to advanced trading strategies, along with professional tips for smarter risk management and coverage of the latest market trends through dynamic courses with new interactive lessons each month. The app also features a real-time trading simulator, where users can practice their strategies using live market data across stocks, forex, commodities, and more, with 24/7 trading available through options, digital currencies, and fiat currencies.

Trading Game also provides a Stock Market Game where users can buy and sell stocks in a realistic simulator, with live price updates, charts, and a stock screener tool to build a profitable portfolio. The app includes a "Pattern Hunter" quiz to improve pattern recognition skills, and the "Copy to Chart" feature allows users to easily apply technical analysis to their charts.

Additionally, the app offers quick reads with visual summaries of key trading and investing books to accelerate learning, along with Trading Battles, where users can compete with friends and AI in short 1v1 trading competitions to test strategies and improve their skills.[25]

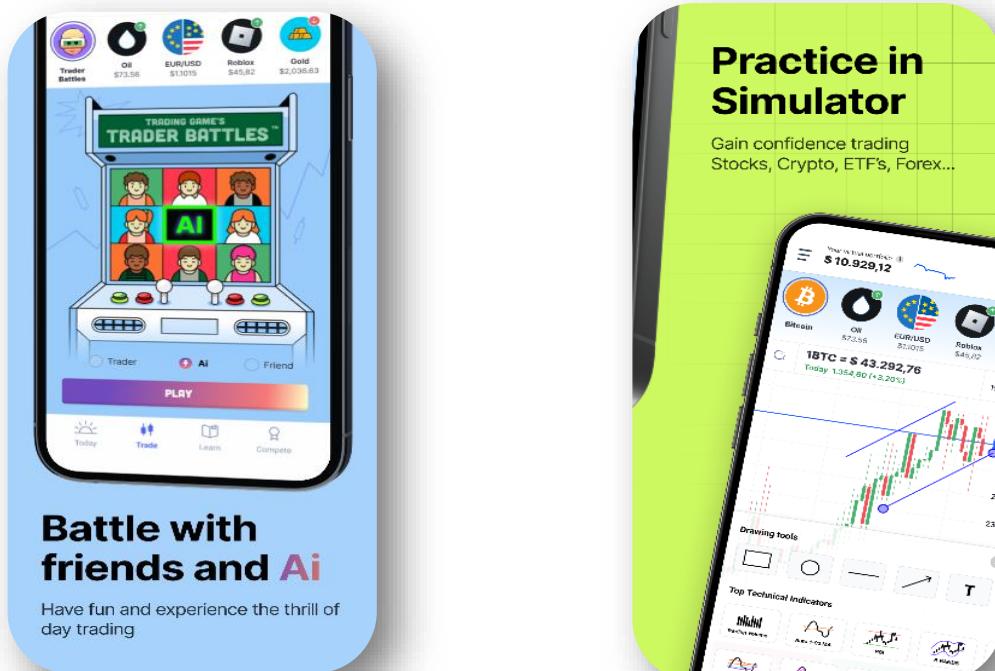


Figure 8: Trading Game Application

4- Stock Trainer: Virtual Trading



The Application is an advanced stock market simulation app that allows users to experience trading using real market data from over 20 global exchanges, including BSE, NSE, NASDAQ, DOW, and S&P. The app is designed to be suitable for both beginners and experienced investors, providing a risk-free environment to test trading strategies with virtual money before investing real capital.

The app offers an exceptional trading experience that mirrors real financial markets, allowing you to trade with live prices and continuous updates from global exchanges. It supports stop-loss and limit orders, enabling you to manage risks effectively. Additionally, it provides comprehensive stock analysis, including historical charts, financial data, and lists of the top gainers and losers, helping you make informed investment decisions.

Furthermore, you can easily manage your portfolio by tracking your performance and adding stocks to your watchlist. The interactive trading simulator offers a complete educational experience, allowing you to develop and refine your strategies based on detailed statistics.[26]

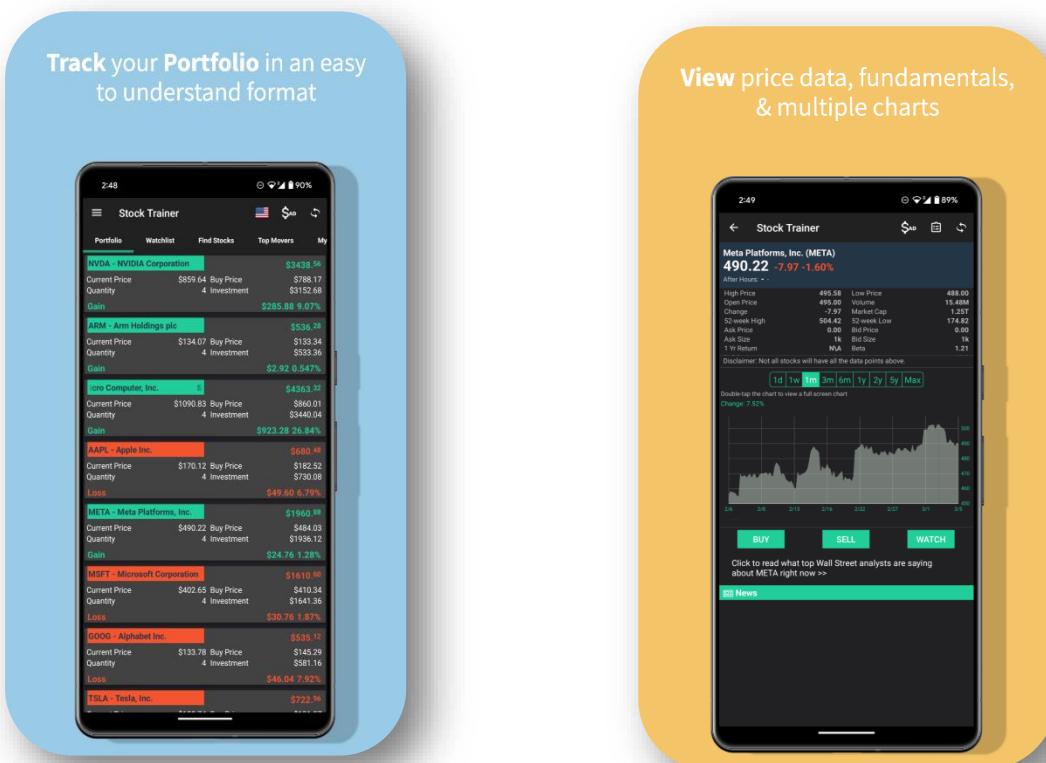


Figure 9: Stock Trainer Application

Semester-2 1446H (Spring 2025)

5- Stock Market Simulator Game



The Application offers a unique opportunity to experience financial market trading using \$100,000 in virtual money, allowing you to test your investment skills without any risks. The app simulates trading in stocks, forex, gold, oil, and indices, with real-time price updates, providing an experience that closely mirrors real-market conditions.

This app delivers a realistic market simulation, enabling you to trade over 200 financial assets with live price updates, making it as close as possible to actual trading. You can test your skills risk-free through a virtual portfolio, allowing you to experiment with different strategies and learn technical analysis and market trends.

Additionally, the app offers an interactive trading course, inspired by top investment books and training programs, to help you effectively learn trading strategies. To make it even more engaging, you can join the weekly leaderboard and compete against other traders to evaluate your performance. If you're interested in forex, the app allows you to explore the secrets of currency trading, including margin trading with different leverage levels, helping you understand the risks and opportunities in this dynamic market.[27]

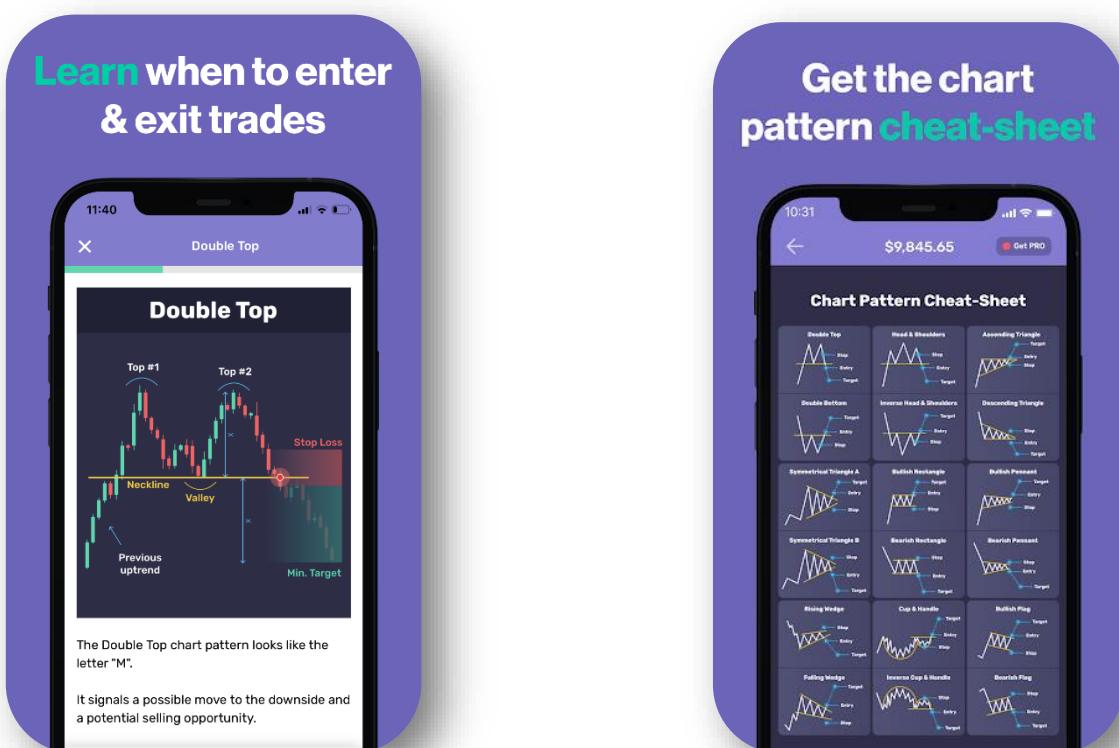


Figure 10: Stock Market Simulator Game application

Semester-2 1446H (Spring 2025)

Table 1: Competitive Product Analysis

Features						
Arabic Language Support	✓	✓	✗	✗	✗	✓
Stock Market Simulation	✓	✓	✓	✓	✓	✓
Virtual Capital	✓	✓	✓	✓	✓	✓
Market News & Financial Analysis	✗	✓	✓	✓	✓	✗
Educational Courses & Tutorials	✗	✓	✓	✓	✓	✓
Interactive challenge	✗	✗	✓	✗	✗	✓
Multi-Market Trading Support	✗	✗	✓	✓	✓	✗
Trading Challenges & Competitions	✗	✗	✓	✗	✓	✓
User-Friendly Interface	✓	✓	✓	✓	✓	✓
Social Trading (Follow & Learn from Investors)	✗	✗	✓	✗	✓	✗
Customization & Preferences	✗	✓	✗	✓	✓	✓
Scenario-Based Learning	✗	✗	✗	✗	✓	✓
Buy & Sell Recommendations	✗	✗	✗	✓	✓	✓
Gamification (Points & Rewards System)	✗	✗	✓	✗	✓	✓
Beginner-Friendly Features	✗	✓	✓	✓	✓	✓
Crypto Trading Support	✗	✗	✓	✗	✓	✗

5.7 Summary of Literature Review:

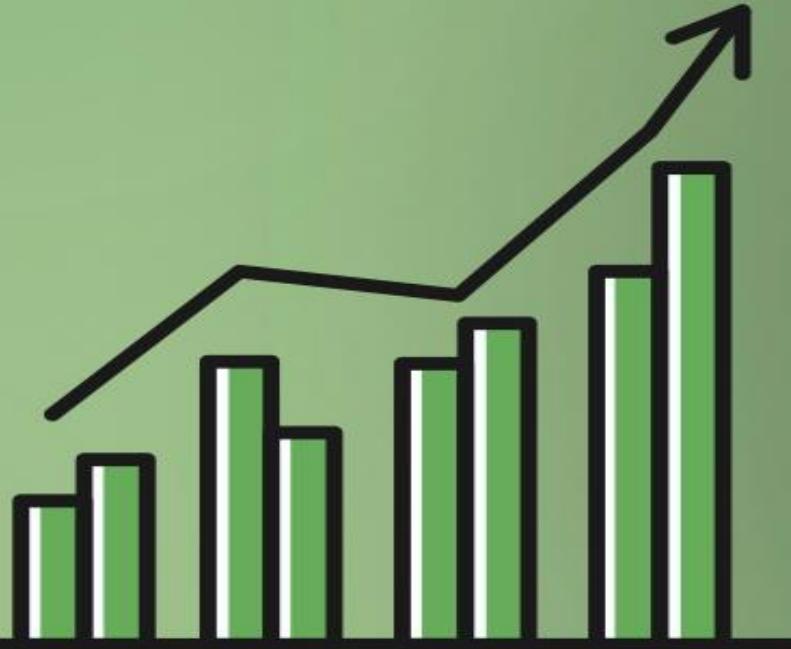
The literature review highlights the effectiveness of stock market simulation and gamification in enhancing financial literacy, decision-making, and user engagement. Several studies emphasize the importance of simulations in providing risk-free environments for practicing trading strategies and improving financial analytical skills [13], [14], [15]. Similarly, other research underscores the role of gamification in motivating users and improving financial behaviors through engaging and interactive learning experiences [16],[17],[18], [19].

Current platforms like Trading Game: Stock Market Sim focus mainly on stock trading simulations, while Stock Market Simulator Game provides a similar environment but lacks engaging interaction. Stock Trainer: Virtual Trading offers a practical stock trading simulation without deep educational content. On the other hand, Investmate delivers traditional educational content but lacks an interactive simulation or engaging challenges. Tadawul Virtual provides a virtual trading environment but does not sufficiently motivate new users. When comparing these applications, most of them lack Arabic-language content.

In contrast, Nomu stands out by integrating a stock market simulator with an interactive and simplified learning experience designed specifically for beginners. Nomu offers engaging challenges, Arabic-language content, and simplified explanations, making investing more accessible to users with no prior experience. It provides a unique balance between education, practice, and entertainment, distinguishing it from other applications in the market.



04 System Design and Development



6 System Design and Development

This section provides an overview of the design and development process of the Nomu application, including the methodology, requirements, system architecture, data design, interfaces, and implementation. It summarizes how the system was planned, modeled, and built to meet user needs and achieve the project's objectives.

6.4 Methodology

The development of the Nomu application was guided by the Agile software development methodology, selected for its flexibility and suitability for projects that require continuous refinement and adaptation. Agile allowed the team to divide the project into short, iterative development cycles in which requirements were analyzed, refined, and implemented gradually. This approach was particularly beneficial for Nomu, which combines educational content, interactive learning, a reward system, Arabic user interfaces, and a financial market simulation built on historical stock data. Through Agile, the team was able to adjust priorities in response to insights gathered from interviews, questionnaires, and academic feedback, ensuring that the evolving system remained aligned with the project's overall vision and user needs. The iterative nature of Agile also supported early testing and verification of each component—such as tutorials, flashcards, the daily challenge system, and portfolio simulation—resulting in improved quality and reduced cumulative errors as development progressed.

To operationalize the Agile methodology in a structured way, the team adopted the Scrum framework as the organizational model guiding the development of *Nomu*. The formal Scrum roles involved in the project—Product Owner, Scrum Master, and Development Team—are presented in Table (5): Scrum Team, and these roles shaped the workflow, prioritization, and coordination throughout the project. The development was divided into iterative sprints, each beginning with planning discussions and concluding with a review of the completed work. Within this structure, the project made use of the familiar Scrum artifacts, including the Product Backlog, the Sprint Backlog, and the Increment, which were maintained and updated continuously as part of the sprint cycle. Likewise, the core Scrum events—such as sprint planning, day-to-day coordination, review sessions, and retrospectives—were followed in a simplified, project-appropriate manner that preserved their intended purpose without unnecessary complexity. This integration of Scrum practices ensured clarity, continuity, and steady progress while maintaining alignment with supervisory expectations and the overall vision of *Nomu*.

Table 2:Scrum Team

Scrum Team	
Product Owner:	Dr. Qatrunnada Alsmail Dr. Yousra Saud Almathami
Developers:	Raghad Ahmed Hassan Asma Alshilash Doaa Aldobai Alhanouf Aldakel Allah Norah Aljedai
Scrum Master (SM):	Dr. Qatrunnada Alsmail Dr. Yousra Saud Almathami
Stakeholders:	The Examiners Committee at King Saud University, along with beginner and hesitant investors with limited budgets.

In practice, the *Nomu* team applied Scrum principles through a recurring development rhythm. Each sprint involved defining tasks clearly, distributing responsibilities according to each member's skills, and progressing through UI development, backend integration, content creation, gamification mechanisms, and simulation features. Weekly meetings with the supervisors served as sprint reviews, enabling the team to validate completed work, receive constructive feedback, and adjust plans for the following cycle. Retrospective discussions helped the team refine its workflow, enhance coordination, and resolve challenges related to design, data integration, and cross-component dependencies. This iterative, feedback-driven process ensured that *Nomu* evolved from conceptual design into a mature, fully functional educational and simulation system.

Supporting this methodological framework, the team used a set of tools that facilitated organization, version control, and collaborative software development, primarily Jira¹ and GitHub². Jira was employed to manage the project through structured documentation of requirements, sprint planning, task distribution, and progress tracking, providing full visibility into the status of *Nomu* throughout all development phases. GitHub served as the core environment for code management, enabling team members to work in parallel on different components without conflicts through the use of branches, pull requests, and systematic code reviews. Together, Jira and GitHub strengthened the implementation of Agile and Scrum by ensuring clear coordination, accurate traceability, and seamless integration of all features added across the project lifecycle.

¹ <https://student-team-iibdsij2.atlassian.net/jira/software/projects/SCRUM/boards/1/backlog>

² <https://github.com/asmaib/GP-Nomu.git>

6.5 System Requirements

This chapter details the essential stages of the requirements engineering process. The process starts with the identification of user characteristics to ensure a thorough understanding of their needs. Requirements were then elicited and documented through techniques such as interviews and questionnaires, providing valuable insights into user expectations. The Architecture section provides a high-level overview of the system's structure. Furthermore, a Use Case Diagram was developed to illustrate the application functionality and the interactions between users and the system.

6.5.1 System Users

Nomu is an educational and simulation application designed to support the Arabic language. It is tailored for individuals who are interested in investing but lack prior knowledge or experience, as well as those who may feel hesitant to invest due to fears of financial loss. The target users of Nomu are expected to come from diverse educational backgrounds, including high school graduates, university students, and individuals with advanced degrees. While users may vary in their technical expertise, they are anticipated to have at least a basic level of familiarity with technology, such as the ability to navigate mobile applications. A general understanding of the Arabic language is sufficient to interact with the application, as it is specifically designed to cater to Arabic-speaking users. To accommodate this wide range of users, the application incorporates an intuitive and user-friendly interface that ensures accessibility for individuals regardless of their technical skills, prioritizing simplicity and ease of use to enable even those with minimal technological proficiency to engage with its features effectively and confidently.

6.5.2 Requirements Elicitation and Analysis

We used two methods for requirements elicitation: interviews and questionnaires. These methods targeted our potential users, enabling us to gain a deep understanding of why our system is needed, the extent of its demand, and what users expect from it. We engaged participants with prior knowledge and experience in investment, as well as those with no experience or knowledge in the field. Additionally, we included individuals with diverse interests, skills, and professions to gather a wide range of perspectives and insights. This comprehensive approach allowed us to identify the potential value of our system and underscored the importance of designing it thoughtfully to address user needs effectively.

- **Interviews**

Interviews are a direct and interactive method for gathering detailed insights from potential users. This approach enables us to explore their needs, preferences, and expectations while uncovering valuable qualitative data. By facilitating one-on-one discussions, interviews provide a deeper understanding of user perspectives, making them an effective tool for requirements elicitation.

We conducted four interviews using a mixed approach, combining in-person and online sessions. Among the participants, three had no prior experience or knowledge of investment, while one had experience in multiple investment fields. The three participants with no prior experience expressed a strong desire to try investing but cited several barriers, including a lack of knowledge and fear of financial loss. All four participants identified increasing income as their primary motivation to learn about investments. Additionally, one participant mentioned a

Semester-2 1446H (Spring 2025)

desire to enhance financial skills and develop strategies for long-term financial growth, while another highlighted diversifying income sources as a key motivation.

When discussing the challenges faced in learning about investments, financial loss was a major concern among all participants. Other challenges mentioned included a lack of understanding of where to begin, the overwhelming diversity of educational platforms, and the difficulty in identifying reliable sources of information. One participant specifically mentioned the fear of fraud and the risks associated with taking expert recommendations without experiencing a simulated investment process.

Regarding preferred learning methods, all participants expressed a preference for videos as an engaging and effective medium for learning. Two participants suggested combining videos with written text for better comprehension and as a way to summarize complex ideas. One participant expressed a preference for live online sessions, emphasizing the value of being able to ask questions and interact with an instructor in real time. Another participant highlighted the need for simple, step-by-step guidance to make learning easier and less intimidating.

None of the participants had previously tried investment simulator applications. However, one participant showed a strong interest in using a simulator, noting that it would help build confidence by allowing them to practice in a safe, risk-free environment before attempting real investments.

Gamification was another feature discussed during the interviews. Most participants agreed that incorporating elements such as progressing through levels, and earning badges, would make the learning process more enjoyable and motivating. One participant suggested that having diverse types of games would help sustain interest and engagement. Additionally, participants mentioned the importance of interactive learning, with one specifically requesting the inclusion of success stories from experienced investors to inspire and provide practical insights.

Finally, some participants highlighted the need for support and guidance. Suggestions included the availability of an expert to answer questions, as well as features like monthly progress reports to track their progress. These insights emphasized the importance of creating a user-friendly, engaging, and supportive platform for investment education. For more details about the interviews, see Appendix A.

• **Questionnaires**

We issued an online questionnaire using Google Forms for requirement elicitation to ensure fast and diverse responses. The questionnaire received 215 responses after being distributed via various social media platforms, enabling us to gather detailed information and insights about the target users. The questionnaire questions, flow, and response graphs are included (see Appendix B). The questionnaire consists of 13 questions, with questions 6, 7, and 8 appearing only if the participant answers 'yes' to question 5, which indicates their interest in investment.

We began with demographic questions to identify the characteristics of our potential stakeholders. The results revealed that 35.8% of respondents were aged between 18 and 24, an age group keen on exploring and building financial experiences. Additionally, 21.9% of

Semester-2 1446H (Spring 2025)

respondents were aged 45 and above, likely looking for multiple income sources, while 16.3% were aged 25 to 34. A smaller proportion of respondents, 12.1%, were aged between 35 and 44, an age group often associated with career advancement and seeking stable financial growth. The rest of the respondents were younger than 18, representing 14% of the data. Regarding gender distribution, the majority of participants were female, accounting for 82.8%, while males constituted 17.2%.

In terms of education levels, most respondents had a bachelor's degree (49.8%), followed by high school graduates (30.7%), while the rest were distributed across master's, PhD, and middle school levels. This diversity highlights the range of educational backgrounds among our potential users.

When asked about their current level of knowledge in investment, 45.1% answered that they had no prior knowledge, while 34.4% identified as beginners, and 17.7% considered themselves intermediate. Only 2.8% of respondents classified themselves as advanced investors (Figure 17). Despite this, a significant majority 92.6% expressed interest in learning about investments, while only 7.4% indicated they were not interested.

Respondents were also asked about the types of investments they were interested in, with 68.3% selecting stocks as their primary choice. Additionally, 52.3% showed interest in real estate, followed by 37.7% in technical services and 42.2% in commodities. This indicates a preference for tangible and technology-based investment opportunities.

When we explored motivations for learning about investment, 80.4% aimed to increase their income, while 52.8% had future plans and saw investment as a key component of achieving their goals. Others were driven by the desire to acquire new financial skills (45.7%), either to manage their money more effectively or to explore new financial opportunities. Additionally, 27.1% expressed curiosity about the investment world, wanting to understand how it works and whether it would be beneficial for them.

However, participants also highlighted several challenges they face in learning about investments. The most significant barrier, cited by 69.3% of respondents, was the fear of financial loss, which made them hesitant to start investing. Other major obstacles included a lack of trustworthy resources (58.8%), making it difficult to find reliable information, limited time to learn (44.2%), and difficulties in understanding complex investment concepts (52.8%), which can be overwhelming for beginners.

Interestingly, 63.3% of participants agreed that an educational platform incorporating investment simulations would encourage them to start investing, as it would provide a risk-free environment to practice and build confidence. Meanwhile, 30.7% were undecided, possibly due to uncertainty about how effective such a platform would be, and only 6% disagreed.

When asked about their preferred learning method, the majority, 74.9%, favoured video-based content, as it offers a more engaging and visually appealing way to grasp investment concepts. In contrast, 20.5% preferred written material, possibly valuing detailed explanations and structured content. The remaining participants expressed a preference for a combination of both, where videos would be used to explain investment concepts clearly, while text-based

Semester-2 1446H (Spring 2025)

content would provide additional details and structured explanations for better comprehension. This suggests that incorporating both formats could be beneficial in catering to different user preferences and learning styles.

In terms of application specifications, 70.2% of respondents preferred a mobile application, while 29.8% favoured a website.

Regarding language preferences, the majority, 88.4%, chose Arabic, while 11.6% preferred English. This strong preference for mobile applications and the need for Arabic language indicates that any financial solution we develop should be mobile-compatible and localized to our target users.

Furthermore, 84.7% of participants believed that gamification features, such as earning badges, levelling up, or unlocking achievements, would motivate them to complete their investment learning journey. This highlights the importance of integrating engaging and interactive elements into the platform design to enhance user experience and retention.

• Conclusion

Through a combination of interviews and questionnaires, we successfully gathered valuable insights from a diverse range of potential users, enabling us to identify key requirements for our system.

The interviews revealed that many participants lacked prior investment experience but had a strong desire to learn. The primary motivation for learning investment was increasing income, while others aimed to enhance their financial knowledge or strengthen their future plans. However, the most significant barriers to learning included fear of financial loss, lack of trustworthy resources, and the complexity of investment concepts. Preferred learning methods varied among users, but videos were the most favoured format, with some participants preferring a combination of videos and written content for a deeper understanding. Additionally, there was a strong interest in interactive learning experiences, with users suggesting live sessions, step-by-step guidance, and success stories from experienced investors to enhance engagement.

The questionnaire results further validated these findings. We observed a high demand for localized, mobile-compatible investment education, as the majority of respondents preferred Arabic as the primary language and mobile applications over websites. Additionally, concerns such as fear of financial loss, lack of trustworthy resources, and difficulty understanding complex investment concepts were major barriers preventing users from engaging with investments. The most popular investment types were stocks, real estate, and commodities.

Furthermore, gamification emerged as a highly requested feature, with participants agreeing that elements such as earning badges, progressing through levels, and unlocking achievements would increase motivation and make the learning experience more engaging. Additionally, many users emphasized the need for a simulation feature, which would allow them to practice

Semester-2 1446H (Spring 2025)

investing in a risk-free environment and build confidence before making real financial decisions.

These findings highlight the importance of designing an interactive, user-friendly, and well-structured investment education application. The results suggest that integrating videos, written content, gamification, and simulation features would effectively reduce users' hesitation and meet their learning needs.

By incorporating these insights from the requirements elicitation phase, we aim to develop Nomu, a comprehensive investment education application that effectively meets user needs by simplifying complex investment concepts, offering a risk-free learning environment, and empowering users to gain financial confidence through engaging and interactive methods.

6.5.3 User Interactions

The use case diagram in Figure 12 illustrates the main interactions between the user and the Nomu mobile application. This system offers a range of educational, simulation, and investment-related features designed to enhance financial literacy and decision-making skills.

In this diagram, users can register an account, log in, and manage their profile, with an optional extension to reset their password if needed. Once authenticated, users can view personalized profile information, receive system notifications, and access a variety of learning materials. The learning experience includes solving challenges and saving favorite topics.

Additionally, users can engage with historical stock market data by viewing stock markets, exploring the top 5 highest-performing stocks, and viewing predicted stock performance. In the simulation environment, users can buy and sell virtual stocks, save favorite stocks, and track their virtual portfolio. The system also allows users to reset their portfolio and manage saved stocks by removing them as needed.

Semester-2 1446H (Spring 2025)

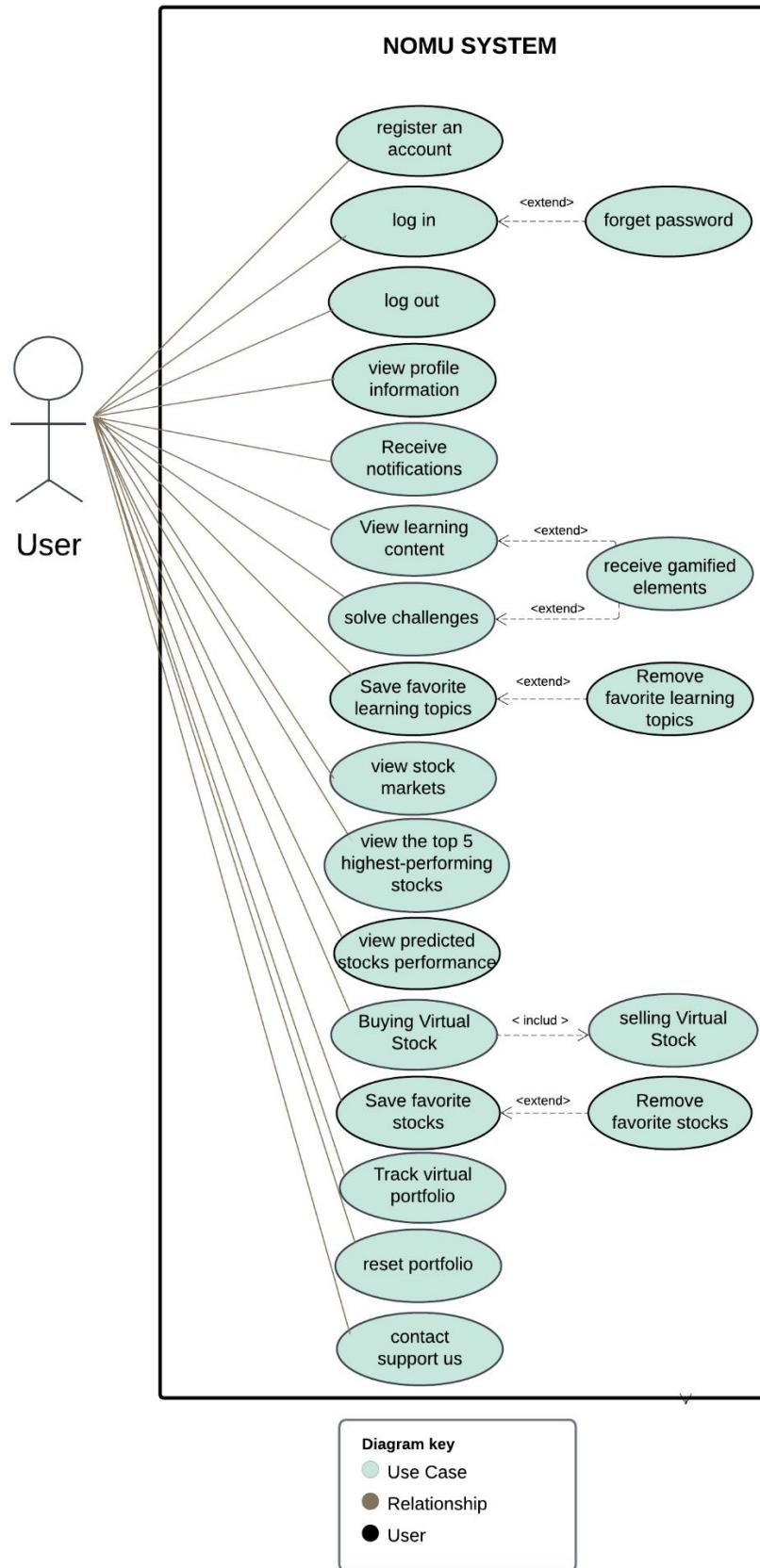


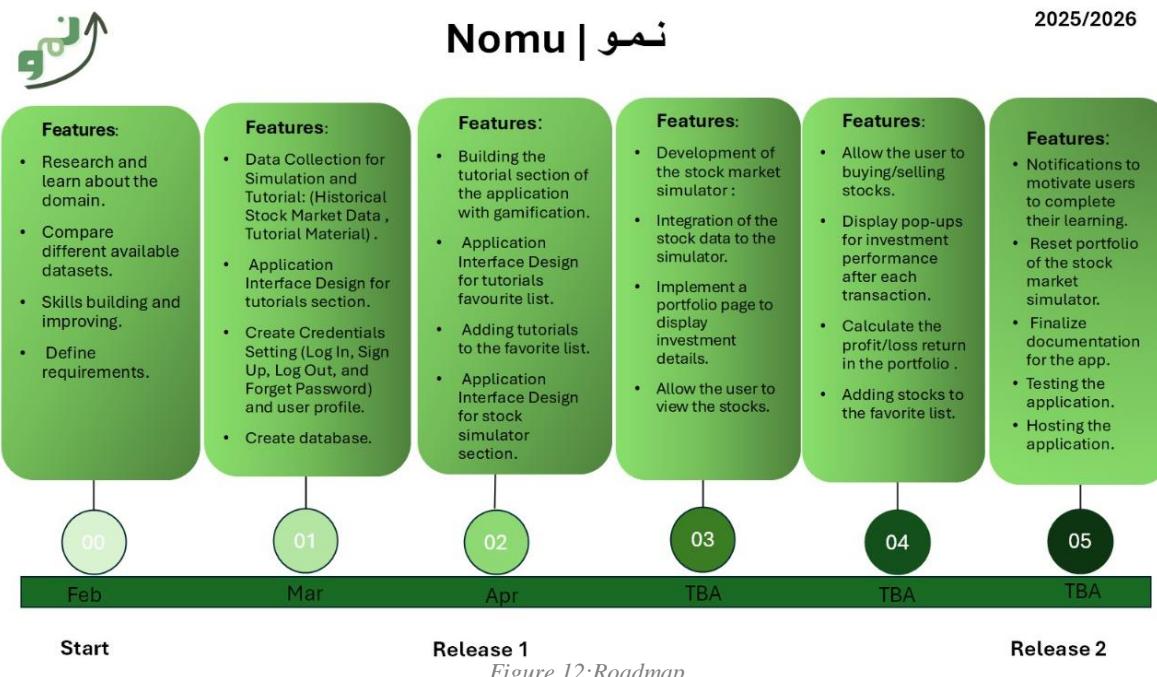
Figure 11: Nomu use case diagram

Semester-2 1446H (Spring 2025)

6.5.4 Roadmap and Product Backlog

- Product Roadmap

In this section, we present the product roadmap for the Nomu application. Figure 12 illustrates the planned development timeline, showing the major phases, feature releases, and progression of work across the project.



- Product Backlog

Table 3 shows the merged list of functional and non-functional requirements for the Nomu application. The table includes all user stories that define the system's functional behavior, along with the non-functional requirements that specify the expected quality attributes of the application. These requirements were gathered and refined through the elicitation process and represent the complete set of needs that guide the development of Nomu.

Table 3:Product Backlog

ID	PBI	Size	Type	Status	Acceptance Criteria
Sprint 1					
1	As a new user, I want to create a new account through "sign up" so that I can access the Nomu application.	2	Feature	DONE	<ul style="list-style-type: none"> As a new user, when I open the application for the first time, then I should see a "Sign Up" button on the start page. As a new user, if I click on the "Sign Up" button, I should be directed to the registration page with fields for Full Name, Email Address, Password, and Confirm Password. If the user left a field empty, then an error message "This field is required." is displayed. As a new user, if I enter valid information (e.g., Full Name: 2-50 characters, +8 characters), I can click the "Sign Up" button to successfully create an account. If the user enters an invalid email format, then an error message "Please enter a valid email address (e.g., name@example.com)" is displayed. If the user clicks "Create Account" and any required fields are empty or invalid, then error messages are displayed for the respective fields and submission will be terminated. As a new user, if I attempt to sign up with an existing email, I should see an error message stating "This email is already registered."
2	As a user, I want to be able to log in with my email and	2	Feature	DONE	<ul style="list-style-type: none"> As a returning user, I should see a "Log In" button on the start page.

Semester-2 1446H (Spring 2025)

	password so that the system can authenticate me, and I can access my account and utilize the services.				<ul style="list-style-type: none"> • As a returning user, If I click "Log In," I should be directed to a login page with fields for Email and Password. • As a returning user, If I enter valid credentials (email and password), I should be logged in and redirected to the homepage. • As a returning user, If I enter invalid credentials, I should see an error message (e.g., "Incorrect email or password."). • As a returning user, If I forget my password, I should see a "Forgot Password" option that directs me to a recovery process. • As a returning user, if I did not enter my email or password and left one or both of them empty then click on "Log In" button, the login process should fail, and an error message should appear • If the user clicks the "Don't have an account?" option, then they are navigated to the Sign-Up Page. • If the user clicks the "Back" button, then they are navigated to the Start Page.
3	As a user, I want the ability to log out of my account , so ensure security of my personal information and maintaining privacy	1	Feature	DONE	<ul style="list-style-type: none"> • As a logged-in user, when I click "Log out" on the user profile page, a confirmation pop-up appears asking the user for confirmation with text: "Are you sure you want to log out?". • As a logged-in user, when I click "Cancel" in the pop-up, the logout is canceled, and I remain on the user profile page.
4	As a user, I want to reset my password if I forgot my password so I can regain access to my account.	2	Feature	DONE	<ul style="list-style-type: none"> • As a user, if I click on "Forgot Password?" on the login page, then a form will appear that will allow me to enter my email. • As a user, if I submit my email address and it exists in the database, a reset link is sent to my email address. • As a user, if the email exists in the database, I am redirected to a page confirming the email has been sent.

- As a user, if I click the reset link that I received in the password reset email, I am redirected to a page to set a new password.
- As a user, if the password is less than 8 characters or lacks complexity, an error message is displayed.
- As a user, if the link is invalid or expired, an error message is displayed.
- As a user, if I click on the "Resend" button, then a new reset link will be sent to my email.
- As a user, if I successfully enter a valid new password and click the "Reset" button, then I should see a confirmation message indicating that my password has been reset, and the new password should be updated in the database.
- As a user, if I successively reset my password and click on "Go back to the Login page" that will appear on confirmation message I should redirect to the log in page.

Sprint 2

			Feature	DONE	
5	As a user, I want to access a library of tutorials so that I can learn about various aspects of stock market investing.	5			<ul style="list-style-type: none"> • As a user, if I navigate to the "Tutorials" section of the app, then I should be presented with a list of available tutorials. • As a user, the tutorial content should be accurate and informative. • As a user if I select a specific tutorial from the list, then the application should load the tutorial content, which may include text, images, videos, and interactive elements. • The tutorial content includes multimedia elements, then the application should ensure smooth and efficient playback of videos and audio. • I interact with the tutorial content (e.g., click on links, answer questions), then the application should provide

Semester-2 1446H (Spring 2025)

					appropriate feedback and navigation to the next steps.
6	As a user, I want the option to challenge myself interactively in the tutorial section, based on my investment knowledge so I can test my skills.	4	Feature	DONE	<ul style="list-style-type: none"> As a user, if I navigate to the tutorial section, I should see an option to challenge myself interactively. As a user, if I choose to take the interactive challenge, I should be presented with a challenge based on the tutorials I've learned. As a user, if I participate in the challenge, I should be able to submit my answers and receive feedback on whether I was correct or not. As a user, if I decide not to take the challenge, I should be able to skip it and return to the tutorial section.
7	As a user, I want to receive gamified elements within the tutorials to enhance my learning experience and engagement.	5	Feature	DONE	<ul style="list-style-type: none"> As a user, if I complete a lesson or module within a tutorial, then I should be awarded points. As a user, I should be able to view my accumulated points in a dedicated section of the app. As a user, the gamification elements should be visually appealing and integrated seamlessly into the tutorial design.
8	As a user, I want to add my favorite tutorial to my favorite list, so I have quick access.	3	Feature	DONE	<ul style="list-style-type: none"> As a user, if I view a tutorial, then I should have the option to "favorite" that lesson. As a user, if I favorite a tutorial, then it should be added to my list of favorite tutorial section in the favorite page. As a user, if I access my list of favorite tutorials, then I should be able to easily view all the tutorials I have added to my favorite page As a user, if I add a tutorial to the favorite tutorials section on my favorite page and click on it, I want to be redirected to the tutorial main page to view its details.

9	As a user, I want to remove a tutorial from my favorites tutorial section so that I can reorganize my favorites list.	3	Feature	DONE	<ul style="list-style-type: none"> As a user, if I have a tutorial saved in my favorites list, then I should have the option to remove it. As a user, if I remove a tutorial from my favorites list, then that tutorial should be removed from the list immediately. As a user, the process of removing a tutorial from my favorites list should be simple and intuitive. As a user, I should receive clear visual feedback after removing a tutorial (e.g., a confirmation message, the tutorial disappearing from the list).
10	As a user I want to ask trading and finance questions to an AI-powered chatbot So that I can learn trading and finance concepts interactively without leaving the app	3	Feature	DONE	<ul style="list-style-type: none"> As a user, when I navigate to the Home or Support screen, then I should see a clearly labeled "Chatbot" icon/button. As a user, if I tap the "Chatbot" icon/button, then the chatbot screen should open showing a text input field and a "Send" button. As a user, if I type a trading or finance question into the input field and tap "Send," then my question should appear immediately in the chat history and a loading indicator should display until a response is received.

Sprint 3

11	As a user, I want to view a list of stocks within the simulator, so that I can explore trading opportunities and make informed investment decisions.	5	Feature	Done	<ul style="list-style-type: none"> As a user, if I access the "Market" or "Stocks" section of the simulator, then the application should display a list of stocks. As a user, if I view the list of stocks, then each stock should be displayed with its symbol, company name, current price, price change (e.g., percentage change, absolute change), and trading volume. As a user, if the list of stocks is displayed, then the application should provide a clear and concise format, making it easy to read and compare stock information. As a user, if I view historical stock data, I should be able to adjust the time frame
----	------------------------------------------------------------------------------------------------------------------------------------------------------	---	---------	------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

for the historical data (e.g., 1 day, 1 week, 1 month).

Sprint 4

12	As a user, I want to buy stock in the simulator so that I can practice investment strategies.	5	Feature	Done	<ul style="list-style-type: none"> As a user, I should have an initial amount of virtual money to start the stock trading simulation. As a user, the simulator should incorporate realistic market conditions, such as price fluctuations, dividends, and market news. As a user, I should be able to access historical market data within the simulator to analyze past stock performance. As a user, if I place an order (buy or sell), then the order should be confirmed with clear details (stock symbol, order type, quantity, price, time of order).
13	As a user, I want to sell stocks using the stock simulator, so I can practice and gain experience in managing my portfolio.	5	Feature	Done	<ul style="list-style-type: none"> As a user, if I own shares of a stock in my simulated portfolio, then I should be able to sell all or a portion of those shares. As a user, the simulator should accurately reflect the proceeds from the sale of my shares in my virtual portfolio balance. As a user, I should be able to choose between selling all shares or a specific number of shares. As a user, I should be able to place a market sell order to sell my shares at the current market price.
14	As a user, I want the application to analyze historical stocks data using AI-driven predictions so that I can receive an informed classification on whether to buy, don't buy a stock	5	Feature	Done	<ul style="list-style-type: none"> As a user, when I navigate to the "Suggestions" page, I should see a list of stocks with their predictions in the future. For each stock, if the prediction indicates that the stock price will go up, I should see a "Buy" suggestion. If the prediction indicates the price will go down, I should see a "don't Buy" suggestion.

					<ul style="list-style-type: none"> As a user, whenever I click on a specific stock from the suggestions list, I should be redirected to the simulation page.
15	As a user, I want to receive a pop-up message after I make a trade using the stock simulator so that I can quickly understand the immediate outcome of my trading decision.	3	Feature	Done	<ul style="list-style-type: none"> As a user, if I complete a trade in the stock simulator (e.g., buy or sell a stock), then a pop-up message should appear immediately. As a user, if I complete a trade in the stock simulator, then the pop-up message should be visually distinct and easily noticeable within the application. As a user, if I complete a trade in the stock simulator, then the pop-up message should clearly indicate whether I made a profit or a loss on the trade, and it should display the amount of profit or loss in a clear and easy-to-understand format (e.g., "+\$100" for profit, "-\$50" for loss). As a user, if I complete a trade in the stock simulator, then the pop-up message should include an option to dismiss it easily (e.g., a close button or a tap outside the message area).
16	As a user, if I add a stock to my favorite stocks section, I want to access it easily so that I can monitor its performance.	3	Feature	Done	<ul style="list-style-type: none"> As a user, if I interact with a stock (e.g., view its details), then I should have the option to "favorite" that stock. As a user, if I add a stock to my favorite list, then it should be displayed in a dedicated "Favorites" section in favorite stock section within the favorite list page. As a user, if I add a stock to my favorites, then the "Favorites stock" section should be easily accessible from the main application screen or through a designated menu option.
17	As a user, I want to remove a stock from my favorites stock section so that I can manage my preferences.	3	Feature	DONE	<ul style="list-style-type: none"> As a user, if I have a stock saved in my favorites list, then I should have the option to remove it. As a user, if I remove a stock from my favorites list, then that stock should be removed from the list immediately. As a user, the process of removing stock from my favorites list should be simple and intuitive.

					<ul style="list-style-type: none"> As a user, I should receive clear visual feedback after removing a stock (e.g., a confirmation message, the stock disappearing from the list).
18	As a user, I want to view a list of the top 5 highest-performing stocks in the stock market simulator, so that I can identify potential investment opportunities and learn about market trends.	4	Feature	DONE	<ul style="list-style-type: none"> As a user, if the system calculates stock performance, then it should identify and rank the top 5 highest-performing stocks based on a defined performance metric (e.g., percentage price increase over a specific period). As a user, if the top 5 performing stocks are determined, then the application should display this information in a clear and user-friendly format (e.g., a table, a leaderboard). As a user, if the market data within the simulator changes, then the list of top 5 performing stocks should be dynamically updated to reflect the latest market conditions. As a user, the list of top 5 performing stocks should be accurately and reliably calculated and displayed.
19	As a user, I want to view the performance of my virtual portfolio so that I can track my investment results and identify areas for improvement.	4	Feature	DONE	<ul style="list-style-type: none"> As a user, I should be able to access a dedicated section to view my portfolio performance. As a user, the application should display my current portfolio value in a clear and easily understandable format (e.g., total value, profit/loss). As a user, the application should calculate and display portfolio performance data in real-time or with minimal latency.
Sprint 5					
20	As a user, I want to reset all my portfolio and trading history, to start over if I make significant mistakes.	3	Feature	DONE	<ul style="list-style-type: none"> As a user, I should have the option to reset my entire simulated portfolio and trading history. As a user, the "reset simulator" option should be clearly labeled and easily accessible (e.g., in the settings menu or within the portfolio overview).

Semester-2 1446H (Spring 2025)

					<ul style="list-style-type: none"> • As a user, before confirming the reset, the application should display a clear warning message explaining the consequences of resetting the simulator (e.g., "This action will delete all your virtual funds, stocks, trading history, and portfolio performance data. Are you sure you want to proceed?"). • As a user, I should be required to confirm the reset action (e.g., by clicking a confirmation button or entering a specific code). • As a user, after confirming the reset, all my simulated portfolio data should be completely deleted. This includes my virtual funds, stock holdings, trading history, and any associated performance statistics. • As a user, after resetting the simulator, I should be returned to the initial state of the simulator with a fresh starting balance of virtual funds.
21	As a user, I want to receive motivational notifications about my stock learning progress, so that I stay encouraged and motivated to continue improving my knowledge of the stock market.	3	Feature	DONE	<ul style="list-style-type: none"> • As a user, if I enable push notifications, then the application should utilize a reliable and efficient notification delivery service. • As a user, the application should handle user device settings for notifications (e.g., "Do Not Disturb" mode) and avoid sending notifications during those times. • As a user, the application should comply with relevant platform-specific notification guidelines (e.g., Apple's Human Interface Guidelines, Google's Material Design guidelines). • As a user, if I enable push notifications within the application settings, then I should receive occasional motivational messages about my stock learning progress.
22	As a user, I want to contact the developers so I can	2	Feature	DONE	<ul style="list-style-type: none"> • As a user, if I navigate to the application profile page and press on "Contact Us"

	give my feedback on the application.				then a developer's email should be appears.
Non-functional requirements					
23	As a user, I want the application's screens to load within 10 seconds under normal network conditions, ensuring quick and efficient user experience. (Performance)	3	Feature	DONE	<ul style="list-style-type: none"> The app should load the main dashboard within 10 seconds on standard devices. Page transitions should occur smoothly without noticeable delays.
24	As a user, I want app icons and labels to be recognizable within 5 seconds , so that I can navigate intuitively. (Usability)	3	Feature	DONE	<ul style="list-style-type: none"> Icon and label recognizability tests with 10 users should confirm a recognition time under 5 seconds for at least 90% of participants. Icons should be clear, intuitive, and designed to convey their purpose visually to users of diverse backgrounds.
25	As a user, I want the system to be available and responsive at least 90% of the time to ensure consistent user access and minimize downtime. (Availability)	3	Feature	DONE	<ul style="list-style-type: none"> As a user, if I try using the system at various times and the system is 95% available without any interruption or downtimes then the system is available.
26	As a user, I want the system to be easy to learn , so that I can understand its features and start using it effectively within 30 minutes. (Usability)	3	Feature	DONE	<ul style="list-style-type: none"> As a user, I expect the system to be considered easy to learn if I achieve an error rate of less than 20% during the training. As a user, I want to be able to utilize the system's features independently after completing a training session within 20 minutes.
27	As a user, I want error messages to be displayed within 5 seconds after	3	Feature	DONE	<ul style="list-style-type: none"> Error messages should appear within 5 seconds in response to invalid input during user testing.

Semester-2 1446H (Spring 2025)

	invalid input, so that I can quickly fix the issue.				<ul style="list-style-type: none"> • Messages should provide actionable guidance for correcting errors. •
28	As a developer, I want the Nomu application to have maintainable architecture , enabling efficient updates, bug fixes, and future development. (Maintainability)	3	Technic al Work	DONE	<ul style="list-style-type: none"> • As a developer, if I want to manage the application using a version control system (e.g., GitHub), then it should be available and help me track changes and collaborate with other developers. • As a developer, if I need to debug the application, I want to locate the issue within 15 minutes • As a developer, I want access to an integrated issue-tracking system (e.g., Jira) to efficiently manage bugs and feature requests, and to prioritize work based on project needs.

Semester-2 1446H (Spring 2025)

6.6 System Design

This section will describe the overall system architecture, design principles, and key components of the Nomu application. It will illustrate how the client-server model supports the application's functionalities, including accessing educational content, simulating investments, interacting with the AI-powered chatbot via the Gemini API, and managing gamified learning progress. Through architectural diagrams, class diagrams, flowcharts, and pseudocode, this section will explain how major modules such as the Learning Module, Daily Challenge, and Video Lesson Rewards are implemented and integrated to deliver a seamless, scalable, and engaging user experience.

6.6.1 Architectural Diagram

The Nomu application employs a client-server architecture to ensure a clear division of responsibilities between the user interface (client-side) and backend processes (server-side). The client-side consists of the mobile application, which provides users with an intuitive platform to explore educational content, simulate investments, interact with a chatbot, and track their progress. The server-side encompasses the backend infrastructure responsible for processing requests and managing data. The two sides communicate through a reliable network, enabling seamless data exchange.

For example, when a user interacts with the application to simulate a stock trade, the client side sends a request with trade details (e.g. stock symbol) to the server. The server then retrieves historical market data from integrated APIs, processes the simulation logic (e.g., calculating profit/loss, updating portfolio value), and saves the results to the database. The server responds to the client with simulation results, such as portfolio updates, performance insights, or visual charts, which are displayed to the user. This process aligns with existing app functionalities, such as authentication, learning progress tracking, and portfolio management, ensuring a seamless and engaging user experience. Similarly, when a user interacts with the Nomu chatbot, the application sends the user's question to the Gemini API using an API key, retrieves the AI-generated response, and displays it back to the user.

This architecture was chosen for several reasons. First, it allows for efficient data management and processing by centralizing computations on the server-side. The server-side utilizes a database management system to store user data, historical market data, and simulation results securely. This reduces the load on user devices, ensuring smooth and responsive application performance even on less powerful devices. Second, it provides a modular framework, enabling us to update, expand, or modify individual components such as the simulation engine or chatbot services independently.

Furthermore, while Nomu does not handle highly sensitive user data, the client-server architecture ensures the accuracy and consistency of educational content and user progress. By centralizing these operations on the server, the system can reliably manage and deliver data, providing users with a stable and dependable learning experience. This approach also streamlines data management, allowing the application to efficiently track user interactions and achievements, ensuring that all progress and updates are accurately reflected across the platform.

Semester-2 1446H (Spring 2025)

In addition to the client and server components, the application integrates with external APIs to retrieve accurate and region-specific market data for simulations, as well as to power the AI-driven chatbot feature via the Gemini API. The client-side leverages these insights to provide an engaging and user-friendly interface using Flutter, while the server-side ensures efficient data handling and processing with Python and relevant libraries like Pandas and Scikit-learn[11].

This architecture ensures that Nomu not only meets current user needs but is also prepared for future growth and feature expansion. By employing a client-server model, Nomu aligns with our vision of delivering a high-quality, engaging, and secure investment education platform.

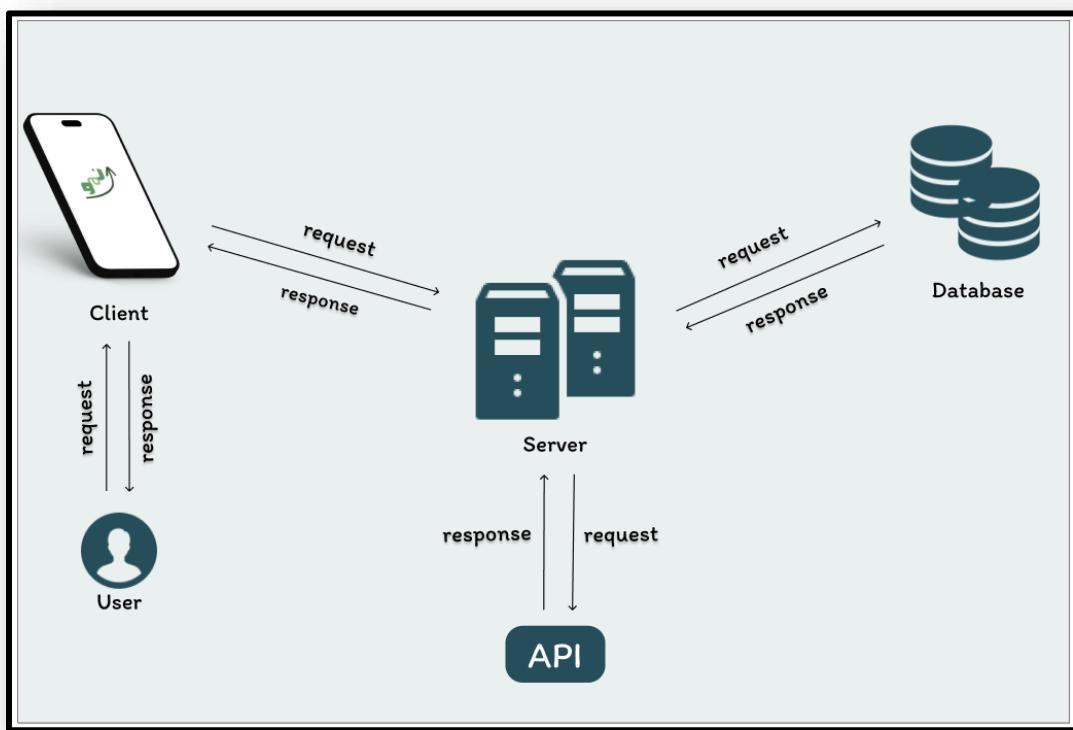


Figure 13: Architectural Diagram

6.7 Class Diagram /DFD

Figure 14 illustrates the class diagram of the Nomu, a mobile application designed to educate users about stock investment through interactive learning, gamification, and simulation features. The system is centered around the User class, which holds essential information such as username, email, walletBalance, and coins. Each user interacts with various modules including daily questions, learning materials, market data, and personal favorite list.

The User class connects several core components:

- **LearningSection** is the primary content hub, storing lesson data like title and description. It is specialized into two types:

Semester-2 1446H (Spring 2025)

- VideoLesson, which includes a video id and url
- PlaylistLesson, which includes multiple videos defined by subtitle and url.
Both provide methods to navigate to related flashcards.
- **Flashcard** represents individual learning cards with a question and answer, stored within each learning section. These cards help reinforce understanding through interactive swiping.
- **DailyQuestion** allows each user to solve one investment-related quiz per day, with attributes such as text, multiple options, and the correctAnswer. The method isCorrect(answer: String): bool is used to verify the user's input.
- **FavoriteList** consolidates all user-specific favorites into a single class. It stores lists for favoriteLessons, favoriteFlashcards, and favoriteStocks, and offers methods such as toggleFavoriteLesson(), toggleFavoriteFlashcard(), and checkers like isLessonFavorite() and isFlashcardFavorite() to manage preferences efficiently.
- **StockMarketData** provides historical market indicators including symbol, price, and volume. This historical data is collected to encourage users, especially beginners, who are our target audience, to explore the world of investment with confidence. By offering simulated trading through fake money, the Nomu application allows users to gain a full investment experience within a safe, risk-free environment. This approach helps bridge the gap between theoretical learning and practical application, making users feel more comfortable and prepared before engaging in real financial markets.

This system decomposition reflects a structured learning path that transitions from foundational understanding through video lessons, to active participation via daily questions and interactive flashcards, and culminates in practical application through stock market simulation. Architecture is designed to reward correct actions with virtual coins, and offer personalized experiences through the favorite list feature, all in service of promoting financial literacy and empowering users to confidently explore investment strategies in gamified, risk-free environment.

Semester-2 1446H (Spring 2025)

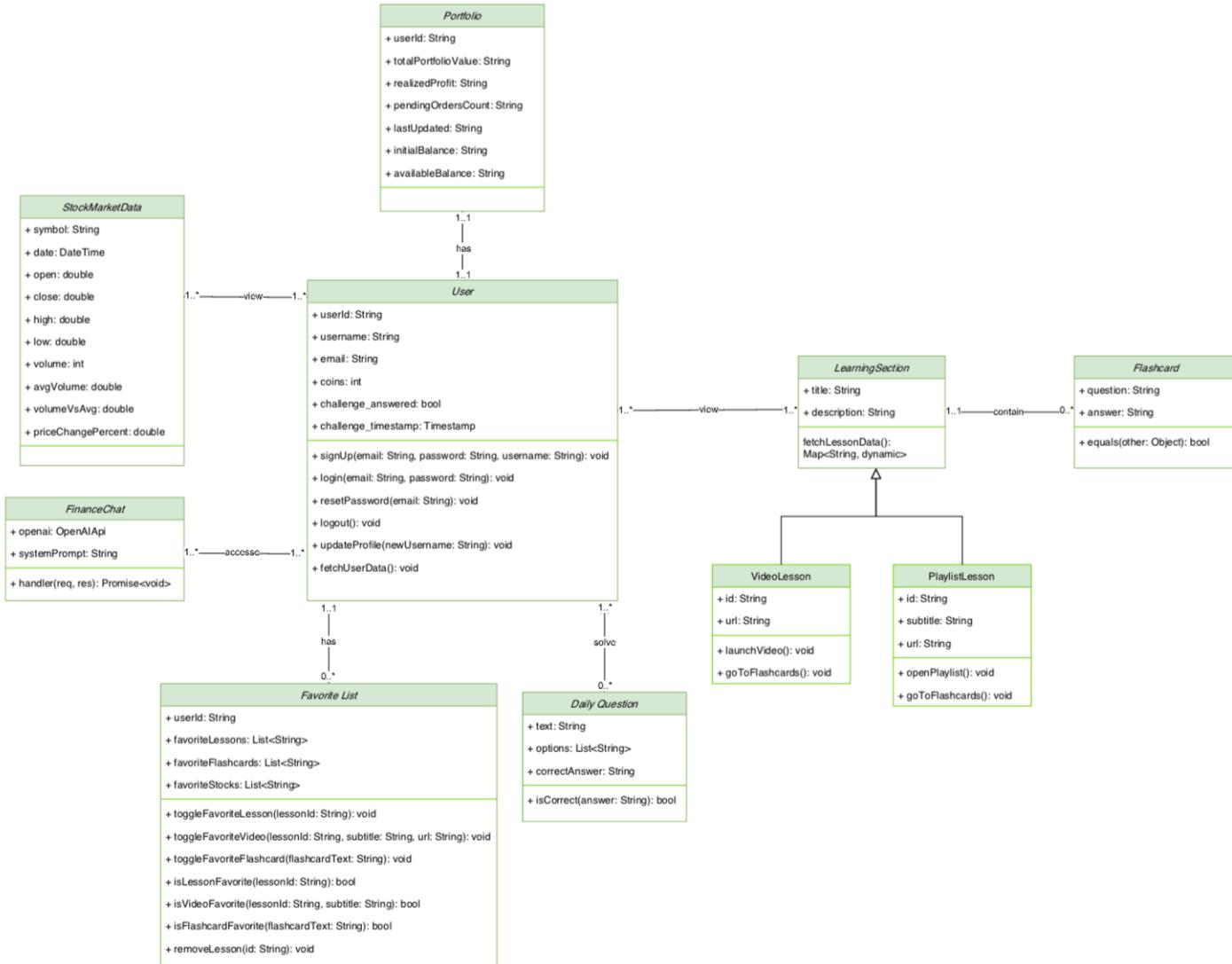


Figure 14: Class diagram

6.8 Component Level Design

This section presents the detailed design of three key components within the Nomu application, focusing on how each is implemented and integrated into the overall system. To illustrate the internal logic and functionality of these components, we provide corresponding pseudocode that outlines their behavior. The selected components are essential to the application's educational and gamification functionalities: accessing the tutorial library, engaging with interactive flashcards, and completing video lessons with reward tracking. Each component is designed to enhance user engagement, support progressive learning, and ensure fair distribution of gamified rewards. Collectively, these components represent the core of the Learning Module, highlighting how system logic is translated into functional features within the codebase.

Semester-2 1446H (Spring 2025)

6.8.1 Access Library of Tutorials

Table 4 shows the user story for Access Library of Tutorials.

Table 4: Access Library of Tutorials

ID	PBI	Size	Type	Acceptance Criteria
5	As a user, I want to access a library of tutorials so that I can learn about various aspects of stock market investing.	5	Feature	<ul style="list-style-type: none"> • As a user, if I navigate to the "Tutorials" section of the app, then I should be presented with a list of available tutorials. • As a user, the tutorial content should be accurate and informative. • As a user if I select a specific tutorial from the list, then the application should load the tutorial content, which may include text, images, videos, and interactive elements. • The tutorial content includes multimedia elements, then the application should ensure smooth and efficient playback of videos and audio. • I interact with the tutorial content (e.g., click on links, answer questions), then the application should provide appropriate feedback and navigation to the next steps.

Definition: Enable the user to explore educational tutorials that cover various investment topics, where each tutorial may include a single video or a playlist, along with associated interactive flashcards.

Pre-condition:

- The user is logged in.

Post-condition:

- The user can view the tutorial content including videos and text.
- The user can navigate to and interact with related flashcards.
- The user can earn coins for watching a video for the first time.

Pseudocode:

```
BEGIN
    FETCH tutorial list from Firestore
    DISPLAY tutorials as cards on LearningPage

    WHEN user taps a tutorial DO
        FETCH lesson document by ID from 'Learing' collection
        DISPLAY title, image, and description

        IF lesson contains 'playlist_videos' THEN
            SHOW description page with "متابعة الدروس" button
            WHEN user taps the button DO
                NAVIGATE to PlaylistPage
                DISPLAY list of video subtitles
                FOR each video DO
                    ON user tap DO
                        PLAY video in VideoPlayerPage
                        LISTEN for video end
                        IF video not rewarded before THEN
                            INCREMENT user's 'coins' by 10
                            MARK video as rewarded
                            SHOW reward dialog
                        ENDIF
                    END
                ENDFOR
            END
        ELSE
            PLAY single video immediately
            LISTEN for video end
            IF not rewarded before THEN
                ADD 10 coins
                MARK video as rewarded
                SHOW reward dialog
            ENDIF
        ENDIF
    END

    IF flashcards available THEN
        SHOW button "انقل إلى البطاقات"
        ON tap, NAVIGATE to FlashcardPage
    ENDIF
END
END
```

6.8.2 Interactive Daily Challenge

Table 5 shows the user story for Interactive Daily Challenge

Table 5: Interactive Daily Challenge

ID	PBI	Size	Type	Acceptance Criteria
6	As a user, I want the option to challenge myself interactively in the tutorial section, based on my investment knowledge so I can test my skills.	4	Feature	<ul style="list-style-type: none"> • As a user, if I navigate to the tutorial section, I should see an option to challenge myself interactively. • As a user, if I choose to take the interactive challenge, I should be presented with a challenge based on the tutorials I've learned. • As a user, if I participate in the challenge, I should be able to submit my answers and receive feedback on whether I was correct or not. • As a user, if I decide not to take the challenge, I should be able to skip it and return to the tutorial section.

Definition: Provide the user with a single multiple-choice investment question each day. The component selects a question at random, allows the user to answer or skip, gives immediate feedback, and if the user answers correctly on their first attempt, coins exactly once per day. The user can also skip the challenge.

Pre-conditions:

- The user must be authenticated.

Post-conditions:

- If the user answers the daily challenge question correctly, they receive 5 coins.

Semester-2 1446H (Spring 2025)

Flowchart:

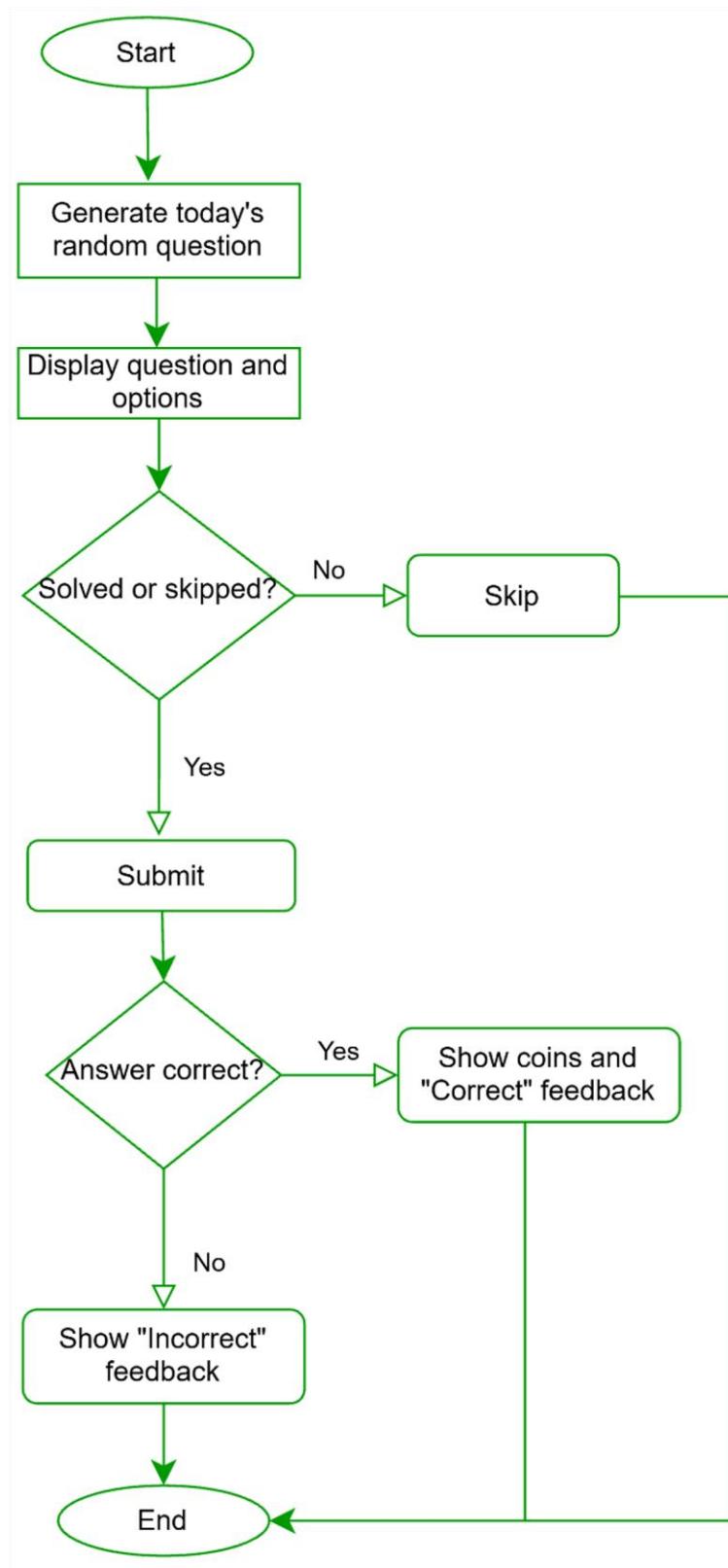


Figure 15: Flowchart of the Interactive Daily Challenge

6.8.3 Video Lesson Completion

Table 6 shows the user story for Video Lesson Completion

Table 6: Video Lesson Completion

ID	PBI	Size	Type	Acceptance Criteria
7	As a user, I want to receive gamified elements within the tutorials to enhance my learning experience and engagement.	5	Feature	<ul style="list-style-type: none"> • As a user, if I complete a lesson or module within a tutorial, then I should be awarded points. • As a user, I should be able to view my accumulated points in a dedicated section of the app. • As a user, the gamification elements should be visually appealing and integrated seamlessly into the tutorial design.

Definition: Reward the user with 10 coins for watching a lesson video to the end — but only once per video.

Pre-condition:

- User must be logged in.
- The selected video must be fully played until the end.

Post-condition:

- User receives 10 coins (if it's the first time watching this video).
- Firestore is updated to record that the reward was already granted.

Semester-2 1446H (Spring 2025)

Flowchart:

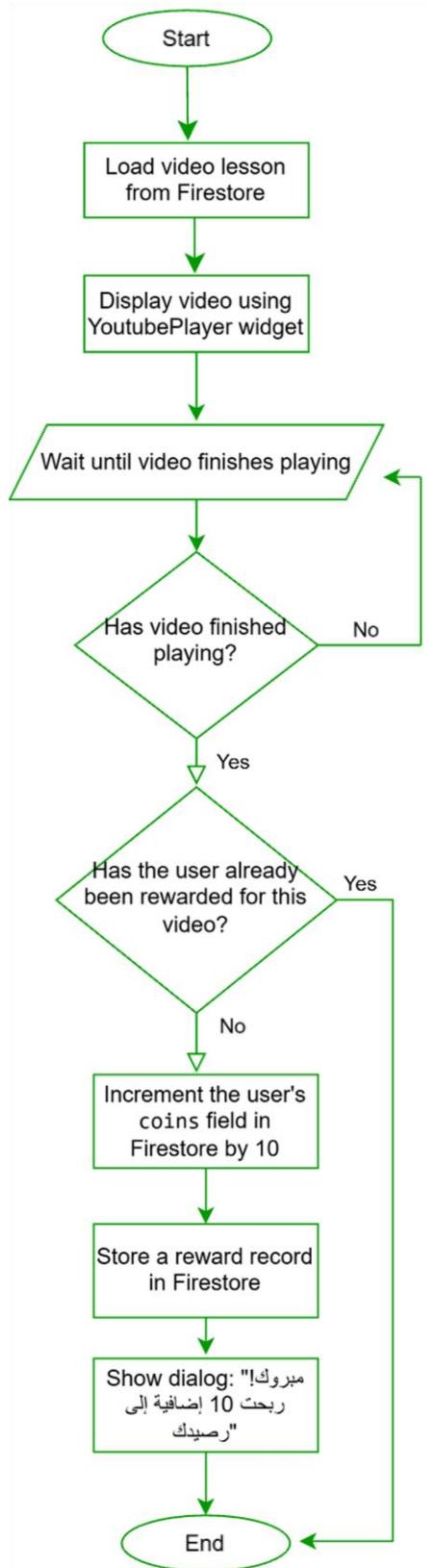


Figure 16: Video Lesson Completion

6.8.4 AI-Based Stock Prediction

Table 7 shows the user story for the AI-driven stock prediction feature.

Table 7: AI-Based Stock Prediction

ID	PBI	Size	Type	Acceptance Criteria
13	As a user, I want the application to analyze historical stocks data using AI-driven predictions so that I can receive an informed classification on whether to buy, don't buy a stock	5	Feature	<ul style="list-style-type: none"> • As a user, when I navigate to the "Suggestions" page, I should see a list of stocks with their predictions in the future. • For each stock, if the prediction indicates that the stock price will go up, I should see a "Buy" suggestion. If the prediction indicates the price will go down, I should see a "don't Buy" suggestion. <p>As a user, whenever I click on a specific stock from the suggestions list, I should be redirected to the simulation page.</p>

Definition: Use AI-driven prediction models to analyze historical stock data and generate a binary classification indicating whether the user should "Buy" or "Don't Buy" a given stock based on short-term predicted trends.

Pre-condition:

- The user must be logged in to access personalized suggestions.
- Historical stock data must be available in the system.
- The prediction model must complete processing before the suggestions page is loaded.

Post-condition:

- The user receives a clear classification ("Buy" or "Don't Buy") for each listed stock.
- The user is successfully redirected to the simulation page upon selecting a recommended stock.
- The system logs the displayed predictions for consistency and future reference.

Pseudocode:

```
BEGIN
    REQUIRE user to be logged in
    IF user not logged in THEN
        NAVIGATE to LoginPage
        EXIT
    ENDIF

    WHEN user opens SuggestionsPage DO
        FETCH list of stocks
        FETCH historical data for each stock
        FETCH predictions for each stock

        INITIALIZE suggestions_list

        FOR each stock DO
            GET predicted_direction for stock

            IF predicted_direction is UP THEN
                SET classification = "Buy"
            ELSE
                SET classification = "Don't Buy"
            ENDIF

            ADD stock with classification to suggestions_list
        ENDFOR

        DISPLAY suggestions_list

    WHEN user selects a stock DO
        NAVIGATE to SimulationPage with selected stock
    END
END
END
```

6.8.5 Virtual Portfolio Performance

Table 8 presents the user story for viewing and tracking the performance of the user's virtual investment portfolio within the Nomu application.

Table 8: Virtual Portfolio Performance

ID	PBI	Size	Type	Acceptance Criteria
19	As a user, I want to view the performance of my virtual portfolio so that I can track my investment results and identify areas for improvement.	4	Feature	<ul style="list-style-type: none"> As a user, I should be able to access a dedicated section to view my portfolio performance. As a user, the application should display my current portfolio value in a clear and easily understandable format (e.g., total value, profit/loss). As a user, the application should calculate and display portfolio performance data in real-time or with minimal latency.

Definition: Provide users with real-time or near real-time visibility into their virtual portfolio performance by aggregating stock positions, calculating profit/loss, updating total value, and presenting the results in a dedicated portfolio section.

Pre-condition:

- The user must be logged in to access their personalized virtual portfolio.
- The system must have stored investment transactions and portfolio data for the user.
- Market prices or simulation values must be available for performance calculations.

Post-condition:

- The user is able to view updated performance metrics (total value, profit/loss).
- The portfolio section reflects the latest simulated stock prices or calculated values.
- The system ensures consistent and accurate updates based on stored transactions.

Semester-2 1446H (Spring 2025)

Flowchart:

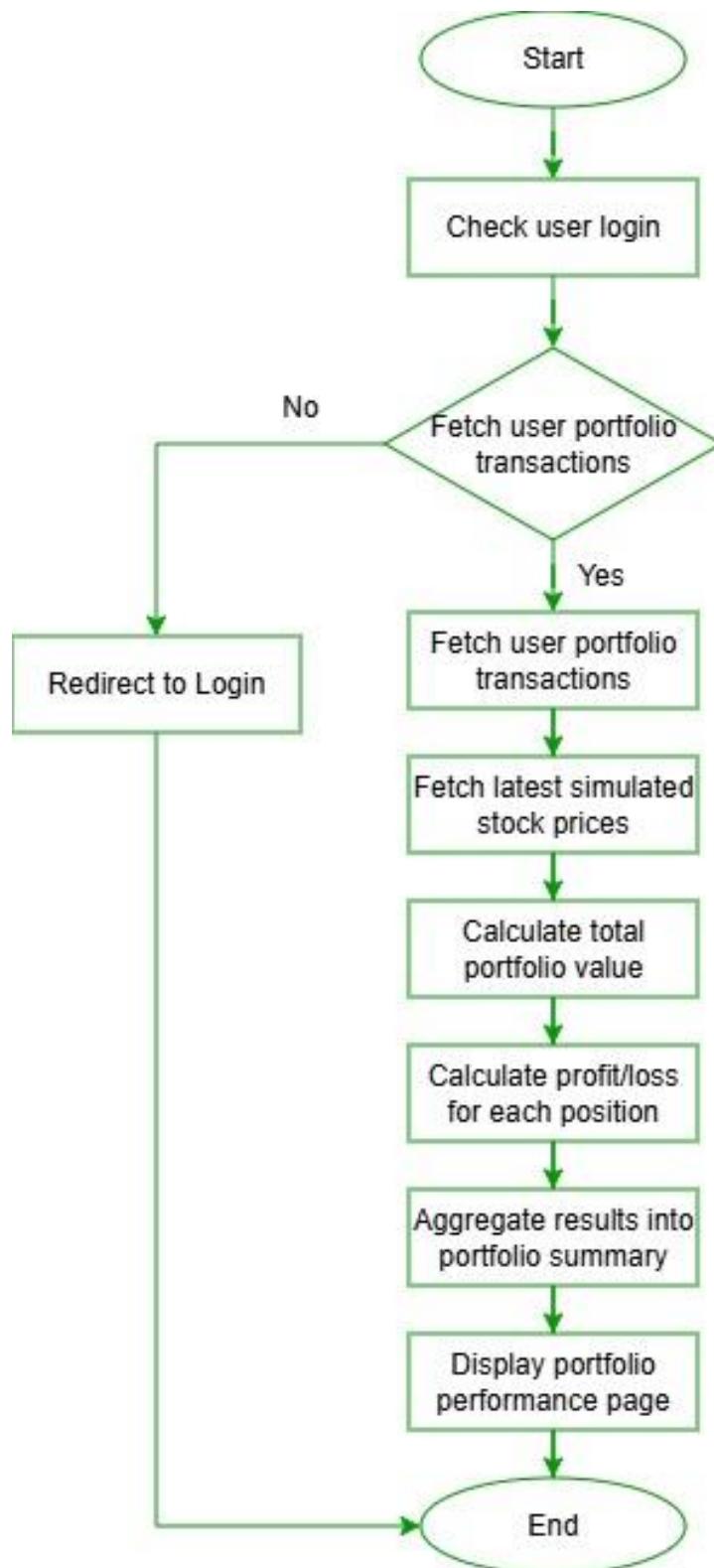


Figure 17: Flowchart Virtual Portfolio Performance

6.9 Data Design

data modeling clarifies how Nomu organizes and links its information. In this section, we present two complementary views of Nomu's data design: a traditional Entity Relationship diagram illustrating the core entities and their relationships, and a Firestore-based non-relational model showing collections, subcollections, and document references.

6.9.1 Data Models

Nomu's system leverages Firebase Firestore as its NoSQL document store. Figure 18 shows the Entity–Relationship diagram, highlighting all core entities and the links between them. To illustrate the actual Firestore structure, we use a tree-style diagram Figure 19 that maps out collections, subcollections, and document fields as they appear in the database.

Nomu ER diagram:

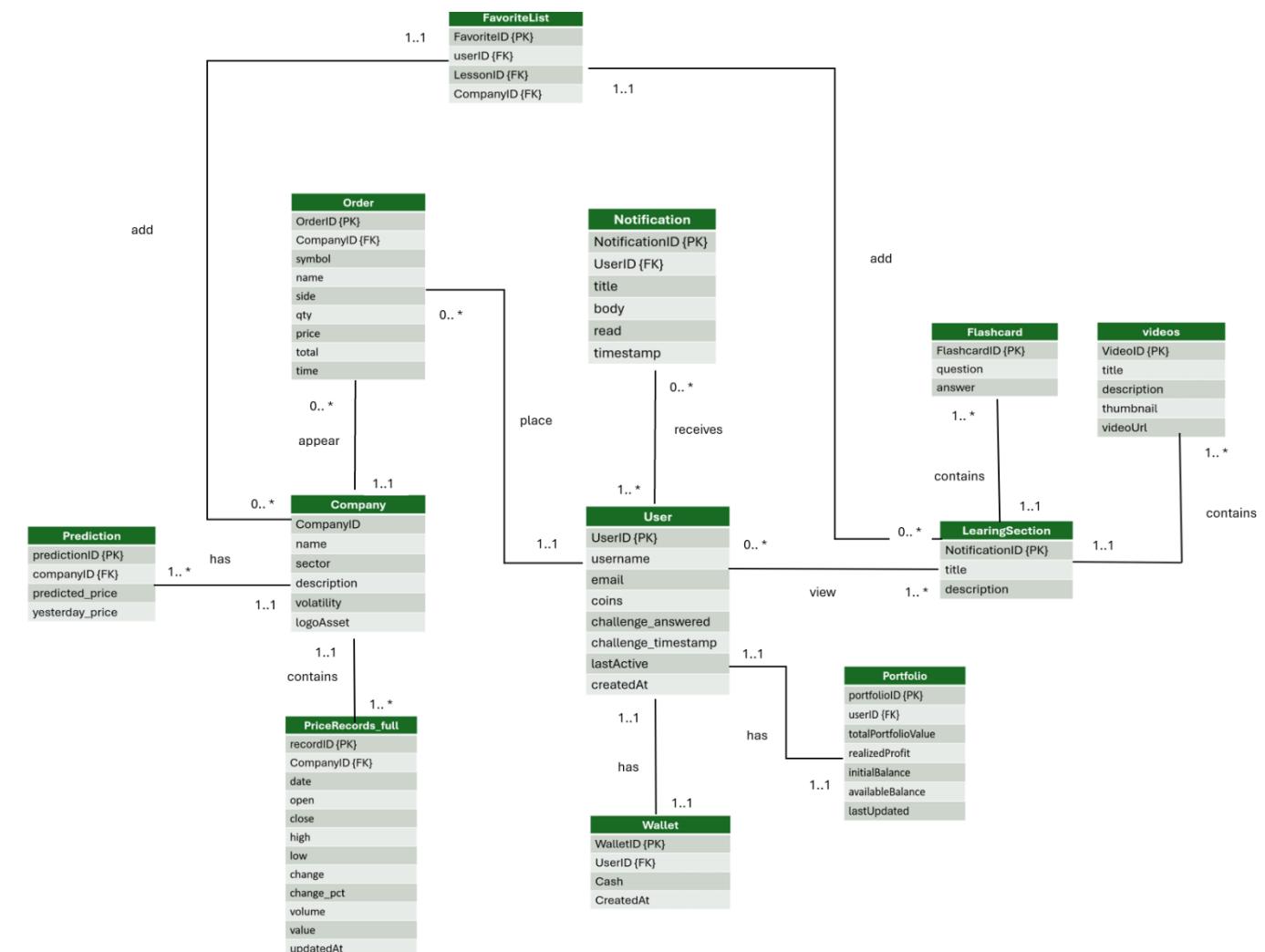


Figure 18: Nomu ER diagram

Non-Relational Data Model

Nomu's Firestore database follows a hierarchical, non-relational structure organized into five main root collections: Users, Learning, Companies, Portfolio, and Favorites. The design combines embedding and referencing to balance performance with scalability.

The Users collection embeds essential profile information such as email, username, coins, challenge answered status, and challenge timestamp. Frequently updated or expanding data is placed into subcollections. Notifications store user messages, orders stores trading actions with full details, and wallet stores balance records. This approach keeps the main user document lightweight while allowing high volume data to grow independently.

The Learning collection embeds lesson metadata including the title, description, and playlist videos. It contains two subcollections that represent learning content. Flashcard stores question and answer pairs, and videos stores tutorial entries with title, description, thumbnail, and URL. This structure reflects the educational hierarchy while supporting independent scaling of each content type.

The Companies collection embeds company metadata such as name, sector, logo asset, and volatility. It also includes two intensive data subcollections. PriceRecords full stores daily historical market information, and market predictions daily stores forecasting values produced by machine learning models. These subcollections isolate large time series data from the main company document.

The Portfolio collection embeds summary statistics including total portfolio value, realized profit, available balance, pending orders count, and last updated timestamp. Each portfolio document references the related user through a userId field, ensuring consistency without duplicating user information.

The Favorites collection stores documents that reference a lesson, flashcard, or stock together with a timestamp. This referencing approach prevents duplication while allowing users to save items of interest across the system.

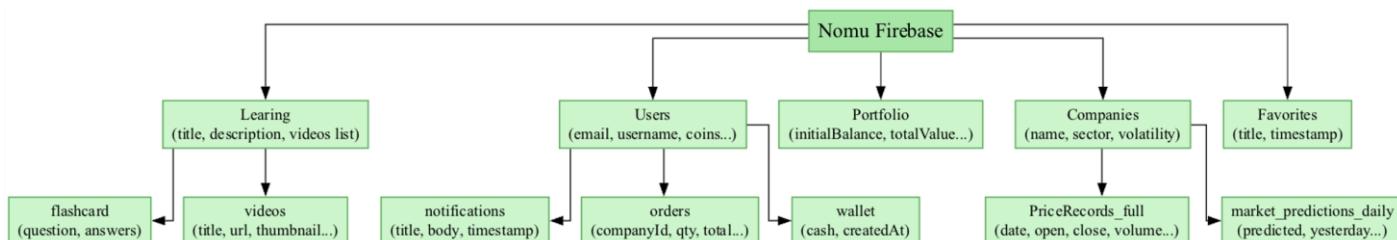


Figure 19: Nomu tree structure

6.9.2 Data Collection and Preparation

In the Nomu project, accurate and reliable data is essential to achieving both of the system's core goals: guiding users through trading simulations and enhancing their investment knowledge. To support these objectives, we collected two main categories of data:

- **Historical stock market data:** used to train predictive models and simulate realistic trading environments.
- **Educational content data:** used to build structured lessons, flashcards, and daily challenges tailored to beginner investors.

For the investment simulation component, we collected real-world historical stock price data from 2015 to 2025 using the Tadawul Reference Data API[10]. This dataset included key features such as stock symbol, date, opening and closing prices, high and low prices, trading volume, and daily percent changes. The collected data serves as the foundation for training machine learning models and running interactive trading simulations within the application.

For the educational component, we curated content directly from Tadawul Saudi Platform, a reputable source of Arabic-language financial knowledge[10]. This included the creation of lesson scripts, flashcards, and daily quiz questions designed to help users understand investment concepts step-by-step. Each lesson was then developed into a simple, interactive video using Canva, designed to simplify complex investment concepts and make the content more accessible to users. To ensure consistent and professional narration in Arabic, we used AI-generated voiceovers from [play.ht](#)[9], a platform specifically chosen for its high-quality Arabic language support. This approach ensured that the educational materials were visually engaging, easy to follow, and professionally delivered.

Stock Data Analysis

To simulate realistic market conditions and train robust predictive models, we collected and analyzed historical stock price data for major Saudi companies. This process encompassed both rigorous data acquisition and extensive exploration to ensure data quality.

For the stock-market simulation and prediction module in Nomu, we relied on real historical price data collected from the Tadawul Saudi stock exchange [10]. The original dataset we purchased covers major Saudi companies from 2015–2025 and was provided as CSV files. For the machine-learning experiments, however, we restricted the training and testing window to 2019–2025 in order to focus on recent market behavior, including the COVID-19 shock, the subsequent recovery, and structural changes in the Saudi market (e.g., Vision 2030-driven reforms). This period is both long enough to train deep models and more representative of the current volatility and trading patterns that Nomu users are likely to face, whereas older data may reflect outdated market regimes, allowing the models to learn from complex patterns of rapid decline and robust growth relevant to the current economic landscape. Each record corresponds to a single trading day and includes the stock symbol, date, open, high, low, close prices, traded volume, and daily percentage change. These files were organized per company and then merged into a unified time-series dataset that feeds both the simulation engine and the machine-learning prediction models.

Semester-2 1446H (Spring 2025)

Before training the models, we applied several preprocessing using the Python programming language[11] and the Pandas library, the dataset was first cleansed of any irregularities by removing rows with missing or clearly erroneous values and by filtering out non-trading days, which could otherwise skew the predictive accuracy. All date fields were parsed and standardized into a uniform format to maintain strict chronological order, a critical requirement for time-series analysis. To facilitate supervised learning, we then engineered additional technical-analysis features such as a 20-day moving average of the closing price, daily returns, and percentage change in closing price. This windowing technique allows the models to analyze sequential price actions over the past month to forecast future movements. For the neural network components, the data was also normalized using Z-score standardization to ensure stability and faster convergence during training. To create supervised labels, we defined next-day targets: for each day t , we looked at the close price on day $t+1$ and encoded the movement simply as "up" or "down" to indicate whether a stock's price would rise or fall on the subsequent trading day.

To determine the optimal engine for the "Market Prediction" page, a comparative analysis was conducted between two distinct machine learning algorithms: a Random Forest Classifier and a Recurrent Neural Network (RNN) utilizing Long Short-Term Memory (LSTM) architecture. The Random Forest model was configured with 300 decision trees to analyze daily features. However, its performance was limited, achieving an accuracy of approximately **0.5263** (based on next-day Close predictions). In contrast, the LSTM model was designed to process normalized sequential data through multiple layers with dropout regularization. Upon evaluation, the RNN (LSTM) model demonstrated superior performance, achieving a high Global Test Metric R^2 of **0.9963** for Close Level predictions. This significant difference highlighted the RNN's ability to capture complex temporal dependencies inherent in stock market fluctuations during this dynamic period that the traditional Random Forest model missed. Consequently, the RNN was selected as the primary algorithm to power the application's prediction features.

Once the RNN model generates its daily insights, the results are integrated into the application ecosystem through a dedicated pipeline. The processed predictions are exported and stored in a specialized Firebase Firestore collection named `market_predictions_daily`. This architecture decouples the heavy computational load of machine learning from the mobile device; instead of running complex models locally, the *Nomu* application simply retrieves these pre-calculated insights in real-time. This ensures that users receive instant, high-accuracy investment recommendations without compromising the application's performance or battery life.

Semester-2 1446H (Spring 2025)

Educational Content Collection

As Nomu is fundamentally an educational application, ensuring pedagogical integrity, accuracy, and cultural relevance of the content was a top priority. The development of educational material was not merely a data entry task but a multi-staged production process involving content curation, linguistic engineering, and multimedia design.

To guarantee authenticity, the core curriculum was synthesized from highly authoritative sources. We derived market regulations and fundamental definitions directly from the Saudi Exchange (Tadawul) Platform [10] to ensure alignment with local financial laws. To provide depth on investment mindset and strategy, we adapted concepts from Benjamin Graham's seminal work, "*The Intelligent Investor*" [28]. These complex financial concepts were then simplified and restructured into scripts tailored specifically for beginner investors in the Saudi market.

To provide a consistent and professional narration, we used AI-generated voiceovers rather than recording manually. We experimented with several AI voice platforms and finally selected [play.ht](#) because it offered the most natural-sounding Arabic voices. For each script, the Arabic text was cleaned and enhanced with manually added diacritics and pronunciation hints so that the AI voice would correctly pronounce financial terms and company names. We then chose a narrator voice that best matched a neutral Saudi dialect and generated the audio. Each audio file was reviewed, and segments with mispronunciations were regenerated and fine-tuned until the speech sounded natural, clear, and suitable for Saudi learners. The final voiceovers were synchronized with the Canva videos during the editing phase, ensuring that on-screen text and narration were tightly aligned.

The flashcards were crafted from the same source materials as the lessons. For each video, we extracted the most important definitions, examples, and “common mistakes” and converted them into question-and-answer flashcards game. This ensured that every card reinforced a specific point from the corresponding lesson instead of introducing unrelated information. The cards were then reviewed by team members to check wording, difficulty level, and consistency of terminology with the videos. Only after this review process were the flashcards uploaded to Firebase and linked to their respective lesson IDs inside the application.

Through this end-to-end workflow and systematic content curation, careful scripting, custom visual design, localized AI voice generation, and the flashcards game for Reinforcement learning, we built a cohesive, professional-grade library of educational content that aligns with the learning goals of the Nomu system and demonstrates substantial development effort.

6.10 Interface Design

The Nomu application is designed to offer users a seamless, engaging, and educational experience in the field of investment. The interface prioritizes clarity, interactivity, and gamification to appeal especially to beginner investors in the Arabic-speaking community. The design follows RTL (Right-to-Left) standards, with consistent styling and intuitive navigation.

6.10.1 Site Map of Nomu

The structure of the application is based on a bottom navigation bar, allowing users to move smoothly between the application core modules. The navigation flow is illustrated in Figure 20, and includes the following:

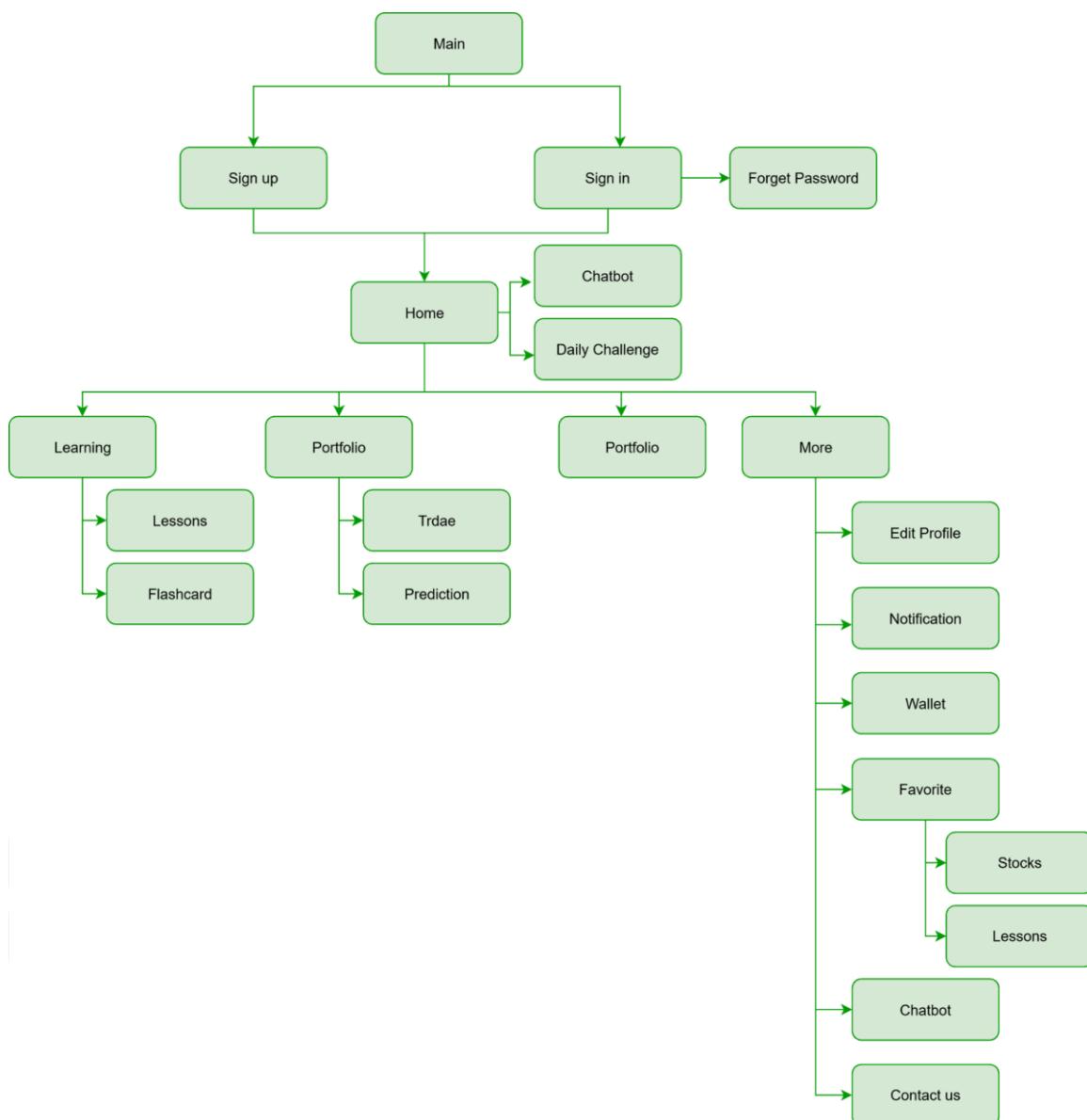


Figure 20:Nomu site map

Figure 20 illustrates the navigation flow of the Nomu application, which is structured around a clean and accessible bottom navigation bar, allowing seamless transitions between its five primary modules:

- **Home Page:** Displays the user's virtual balance, cumulative profit/loss, daily investment challenge (a one-question quiz with coin rewards; also accessible from the Learning Page), and recent trading activity. The page also provides quick access to Nomu's smart assistant (chatbot) and the notification center, where users receive motivational reminders.
- **Learning Page:** Hosts investment educational content created by the Nomu team, curated from the Tadawul Saudi Stock Exchange platform. This section includes:
 - Access to the daily investment challenge
 - A grid of lessons (some with single videos, others as playlists)
 - Flashcards, which are accessible both from the top of the page and within each lesson
- **Simulation Page:** Provides a risk-free environment to practice trading using virtual currency. Key features include:
 - Interactive charts using historical stock data
 - Company-specific descriptions and past performance insights
 - Simulated Buy/Sell functionality
- **Market Prediction Page:** Displays predictions for selected stocks to support learning about market trends. It presents a list of companies with their current price, predicted movement, and a clear "Buy / Don't Buy" recommendation.
- **Portfolio Page:** Visualizes user progress and virtual investment performance. Features include:
 - **Interactive performance chart** with multiple time windows (1D, 1W, 1M, 3M).
 - **Account overview** section showing available balance, total losses, net assets, and cumulative profit.
 - **List of current simulated holdings**, displaying company name, quantity, current price, daily change, and a quick option to mark stocks as favorites.
- **More Page:** Provides additional user utilities and settings, including:
 - Edit personal information (e.g., name)
 - Notification settings to manage app alerts
 - Favorites list for saved lessons, flashcards, and stocks
 - "Talk with Nomu" – access to the in-app assistant

Semester-2 1446H (Spring 2025)

- Wallet to view the user's balance and related info
- Support Center for help and inquiries
- Logout option

6.10.2 Implemented UX Guidelines

The interface was developed in alignment with key UX principles to ensure clarity, consistency, and engagement:

1. Usability and Ease of Use

The application onboarding process introduces its features through a three-screen intro. Lessons and challenges include instructions and labeled buttons like “ابداً للعب” to simplify use. Whether it's taking a daily quiz or flipping flashcards, the experience is streamlined and beginner friendly.

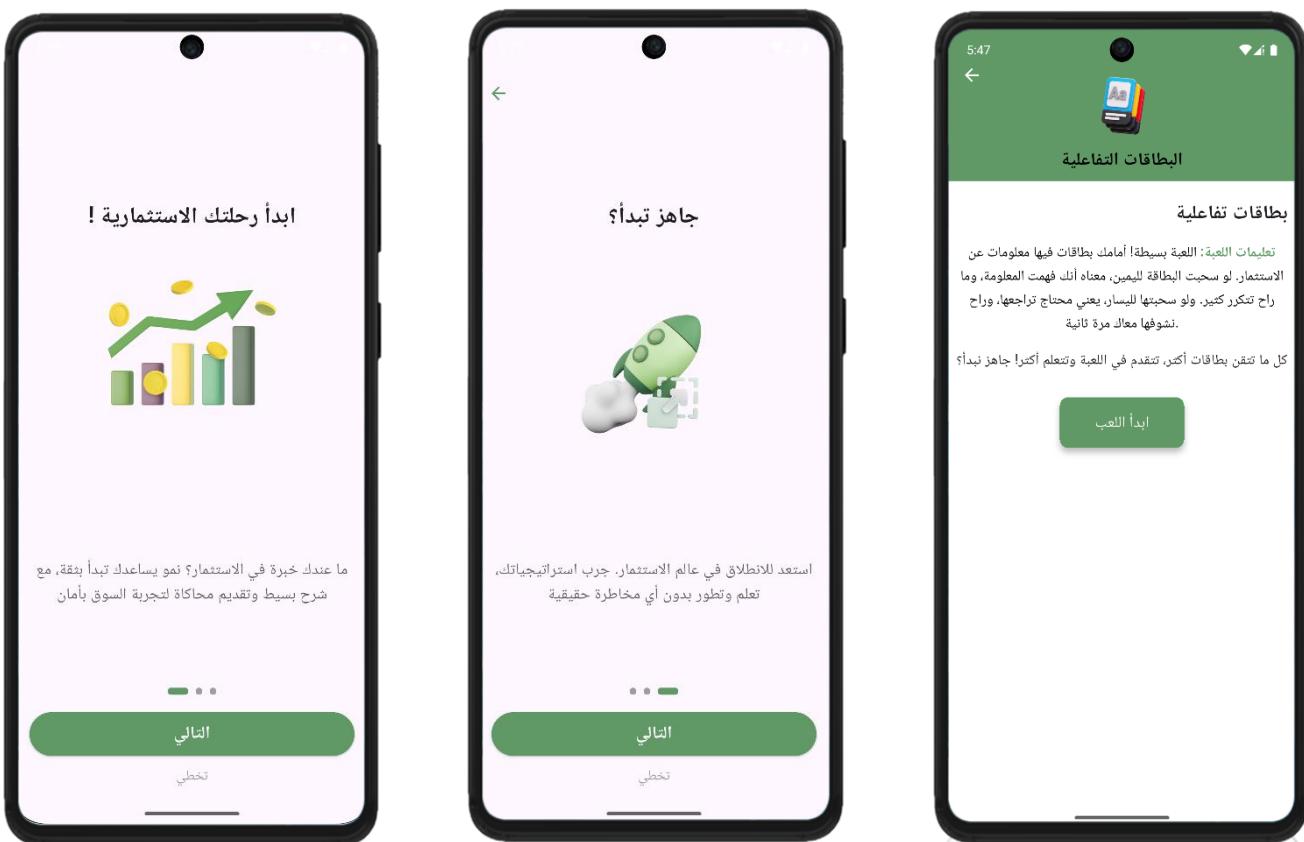


Figure 21: Usability and Ease of Use UX Guideline

Semester-2 1446H (Spring 2025)

2. Clear Navigation

Nomu's bottom navigation bar Figures 22 provides fast access to the application major modules. Each section is intuitively organized. For example, flashcards are available both at the top of the Learning Page and within each lesson, ensuring flexible entry points for users.



Figure 22: Clear navigation UX Guideline

Semester-2 1446H (Spring 2025)

3. Feedback and Confirmation

User actions trigger immediate confirmation, such as dialogs for account creation and password resets. During flashcard sessions, users receive instant visual feedback: correct answers turn green, incorrect ones red. At the end, users are presented with a reward animation.



Figure 23:Feedback and Confirmation UX Guideline

4. Consistency and Standards

Visual consistency is maintained through uniform fonts, rounded corners, and a cohesive green-white color scheme. Input fields, buttons, and navigation icons behave similarly across all screens, reducing the learning curve and improving trust.



Figure 24: Consistency and Standards

Semester-2 1446H (Spring 2025)

5. Smart Validation and Security

Nomu uses real-time validation in all forms to reduce errors and guide users, for example, showing instant messages for invalid emails or short names. Password fields follow strict security rules (12+ characters, uppercase, lowercase, number, symbol), with live feedback as users type, adhering to guidelines recommended by the National Institute of Standards and Technology (NIST) [29].

According to NIST Special Publication 800-63B, longer passwords significantly enhance account security by increasing resistance against brute-force and dictionary attacks, making a 12-character minimum preferable over shorter standards such as 8 characters (Grassi et al., 2017). Additionally, in the password reset screen Figure 25, the success message confirms that a reset link is sent only if the email is valid, highlighting the application's emphasis on user protection and account security.



Figure 25:Smart Validation and Security

Semester-2 1446H (Spring 2025)

6.11 Gamification and Reward Mechanics

To encourage user engagement, Nomu incorporates gamified mechanics tied to learning progress:

- Daily Challenge: A single investment question shown once per day. A correct answer earns the user a coin. Accessible from both the Home and Learning Pages.

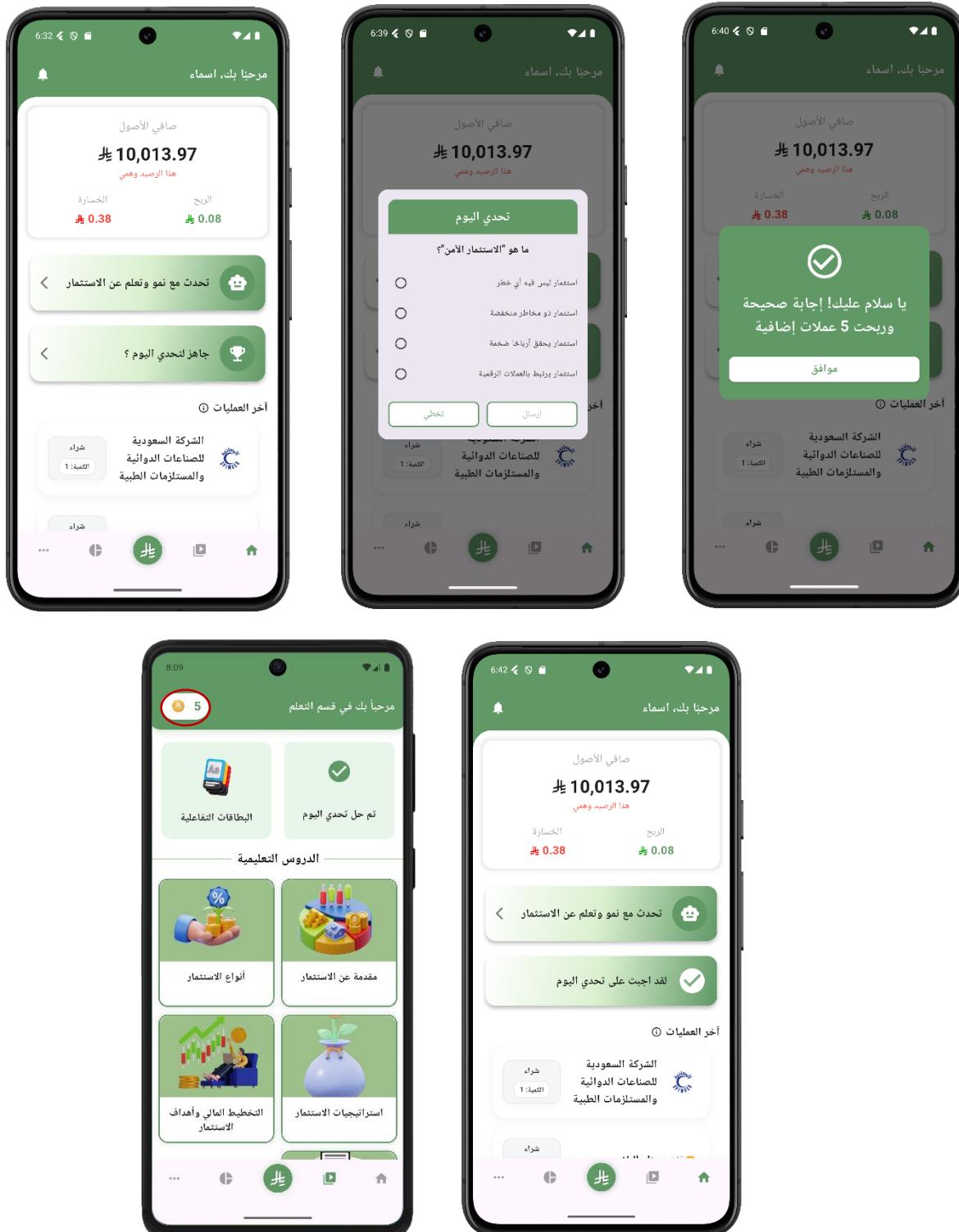


Figure 26:Daily Challenge interfaces

Semester-2 1446H (Spring 2025)

- Flashcards: If a user swipes a flashcard to the right (indicating they know the answer), they earn a coin — but only on their first attempt per lesson.



Figure 27: Flashcard Interfaces

Semester-2 1446H (Spring 2025)



Figure 28: Second Trial reward screen

- Video Lessons: Watching a lesson video for the first time also rewards the user with a coin. Rewatching the same content does not give additional rewards.



Semester-2 1446H (Spring 2025)



Figure 29: Video lessons interfaces

This controlled reward system motivates users to explore new material without encouraging repetition for coin farming.



06 System

Implementation



7 System Implementation

This section provides an in-depth overview of the technical processes involved in the development and implementation of Nomu. It details the key steps and implementation procedures, software components and their configurations, and connections to various data sources and APIs including Firebase (Authentication, Firestore, Storage), YouTube playback, and market data feeds. Additionally, it highlights the main challenges faced during development and the most critical parts of the codebase. Finally, a link to the project's GitHub repository is included for further reference.

7.4 Key Implementation Steps

To ensure Nomu effectively served Arabic-speaking novice investors looking to boost their financial knowledge and confidence, we adopted a clear, phased development approach. The following summarizes the principal stages of its implementation:

1. Requirements Capture

In the initial phase, we conducted user interviews and examined competitor applications to find the essential features that novice investors need. From these discussions, we distilled a core set of requirements: seamless account management (registration, authentication, profile edits), video-based tutorials in both standalone and series formats, interactive flashcards for reinforcement, a daily multiple-choice challenge, a point-based reward system, and the ability to bookmark content. Additionally, a full simulation environment with buy/sell trading actions, AI-based market predictions, a virtual portfolio and wallet system, and support for notifications. These insights shaped our data model and informed us every subsequent design and development decision.

2. Designing the User Interface (UI)

Using Figma, we crafted high-fidelity mockups that adhere to right-to-left reading conventions and established a unified visual language. A persistent five-tab navigation bar was defined to grant instant access to the home, learning, simulation, portfolio, and More screens. We standardized colors, typography, and card layouts across all key components, including educational content,

Semester-2 1446H (Spring 2025)

daily challenges, trading interfaces, market charts, prediction panels, and portfolio overviews to achieve a consistent and polished user experience on different screen sizes.

3. Backend Configuration

On the server side, Firebase Authentication was configured to manage all aspects of user identity—including signup, login, logout, password resets, and profile updates—while Firestore served as the main data store for the application. We structured collections and subcollections for users, lessons, flashcards, earned-coin records, favorites, historical market prices, prediction results, trading orders, portfolio summaries, wallet balances, and notifications. Security rules were implemented to restrict access based on authentication state and indexed queries were added to optimize reads for charts, portfolio reconstruction, and prediction lookups. Offline persistence was enabled to maintain usability and data consistency even under unstable network conditions.

4. Client-Side Assembly

Using Flutter, we built modular screens and widgets to implement the UI prototypes. The onboarding and authentication flows include form validation and smooth RTL transitions. Video tutorials are streamed through the youtube_player_flutter plugin, with lesson URLs retrieved from Firestore. Flashcards use the swipe_cards and flip_card packages to enable a two-pass review experience with color-coded feedback. Daily challenges are selected randomly, stored per user in shared_preferences, and displayed through a custom dialog with Arabic radio buttons. A centralized CoinManager grants points exactly once per learning activity and records them in Firestore, while a dedicated Favorites module allows users to bookmark lessons, videos, or cards.

To support the investment features, we integrated historical market data across multiple modules. The simulation system includes a Market Overview Chart with dynamic filtering, CompanyChart screens with buy/sell actions, and a TradingService that records transactions and updates wallet balances. The prediction module displays actual versus predicted prices with clear recommendations. Portfolio logic reconstructs daily performance by replaying the user's trades, while the Wallet module manages balance resets and synchronization with Firestore. Local reminders and external FCM notifications are handled through a unified notification service that

Semester-2 1446H (Spring 2025)

manages permissions, scheduling, and message delivery. Together, these components form a cohesive, responsive, and interactive experience aligned with the project's educational goals.

5. Quality Assurance and System Testing

We applied multiple layers of testing to ensure the reliability of all learning and simulation features. Unit tests validated key logic such as video loading, flashcard progression, daily challenge selection, coin awards, trading execution, wallet updates, portfolio calculations, prediction comparisons, and Firestore data handling. Widget tests were used to verify essential user interactions across lessons, playlists, flashcards, challenges, market charts, and trading screens.

After component-level validation, we performed comprehensive system testing to confirm correct end-to-end behavior across authentication, video learning, flashcards, daily challenges, simulation charts, prediction model, trading engine, portfolio tracking, wallet management, favorites, and notification services. These tests ensured accurate data flow, stable chart performance, consistent historical-data synchronization, and reliable delivery and logging of external FCM notifications. This combined testing process confirmed that the application operates smoothly and consistently across all features.

7.5 Key Implementation Procedures

This section details the core procedures used to integrate Nomu with its various services and tools, ensuring seamless functionality. It covers setting up Firebase for Authentication, Firestore, and Storage; integrating YouTube tutorials, fetching flashcards and lessons data from Firestore; and persisting each user's daily challenge. Historical and live market data are sourced directly from the Tadawul website through an official data purchase. Finally, it describes the end-to-end workflows, from user sign-in and video playback to flashcard review, daily challenge execution, and coin reward distribution, designed to optimize the learning experience. By examining these technical configurations and design choices, this section reveals how Nomu delivers its interactive, robust feature set.

1. Firebase Setup and Connection

We integrated our Flutter application "Nomu" with Firebase to utilize several key services such as Firestore (for data storage) and Firebase Authentication (for user login/sign-up). This connection was essential to create an interactive and data-driven user experience. The integration process followed these main steps:

1. **Creating a Firebase Project:** We started by creating a project in Firebase through the Firebase console and registered the app using its package name.
2. **Setting up Authentication and Storage:** We enabled Firebase Authentication for managing user accounts and Firestore for storing data in real-time.
3. **Adding Dependencies:** We added the necessary dependencies such as `firebase_core`, `firebase_auth`, and `cloud_firestore` to the `pubspec.yaml` file to access Firebase services.

```
firebase_core: ^3.12.0
firebase_auth: ^5.5.0
cloud_firestore: ^5.6.4
firebase_dynamic_links: ^6.1.3
```

Figure 30 Firebase Dependencies

4. **Initializing Firebase in the App:** We used `Firebase.initializeApp()` in the `main.dart` file to connect the app to Firebase.

```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(MyApp());
}
```

Figure 31 Initializing Firebase

Semester-2 1446H (Spring 2025)

5. Setting up Firestore Data:

We created a data structure and stored it in Firestore, including user account information such as name, email along with coin balance, flashcards, and YouTube videos from the "Nomu" channel. We also stored lesson information, user wallet details, financial balances, stock data, and the favorites list. The data is organized to ensure easy access and real-time updates.

This way, we successfully integrated Firebase into the "Nomu" app and provided reliable and secure services for the users.

2. Credentials Management (Login, Sign-Up, Logout, Password Reset, Profile Editing)

▪ Sign-Up

In **Nomu**, we designed a new registration page, named `signup_page`. The page includes fields such as name, email, password, and password confirmation, along with validation to ensure the data entered is correct. For example, we ensure the password meets security requirements, such as having a mix of uppercase and lowercase letters, numbers, and special characters, and we also validate the email format.

We also added clear error messages to guide the user if there's an issue with the entered data. If the data is correct, the account is created using **Firebase Authentication**,

```
UserCredential userCredential = await _auth.createUserWithEmailAndPassword(  
    email: email,  
    password: password,  
)
```

Figure 32 Sign-Up page

and the user's data is stored in **Firebase Firestore** within a "`users`" collection, where information such as name, email, coin count, challenge status, and default challenge date are stored.

Semester-2 1446H (Spring 2025)

```
await FirebaseFirestore.instance
    .collection('users')
    .doc(userCredential.user!.uid)
    .set({
        'username': username,
        'email': email,
        'coins': 0,
        'challenge_answered': false,
        'challenge_timestamp': Timestamp.fromDate(defaultTimestamp),
    });
}
```

Figure 33 Sign-Up page 2

Additionally, we provided an option for users with existing accounts to log in easily through a dedicated button, allowing smooth navigation between the registration and login pages. Our goal with "Nomu" is to simplify the registration process and provide a secure and user-friendly environment for new users.

▪ Login

The login page in the Nomu app allows users to securely sign in using Firebase Authentication. The page includes the following:

- **Input fields** for email and password with data validation.
- A **login button** triggers the `signInWithEmailAndPassword` function provided by Firebase.

```
try {
    await _auth.signInWithEmailAndPassword(email: email, password: password);
}
```

Figure 34 Login page 1

- A **link to the password recovery page** to help users if they forget their password.
- A **link to the sign-up page** allowing new users to create an account.

Upon successful login, the user is automatically redirected to the **HomePage**.

```
try {
    await _auth.signInWithEmailAndPassword(email: email, password: password);

    Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => HomePage()),
    );
}
```

Figure 35 Login page 2

Semester-2 1446H (Spring 2025)

If there is an error (e.g., incorrect email or password), a custom alert dialog appears in Arabic with a message explaining the issue.

```
try {
    on FirebaseAuthException catch (e) {
        _showMessageDialog(
            'خطا',
            '، عفواً، البيانات المدخلة غير صحيحة. يرجى التأكد من صحة البريد الإلكتروني وكلمة المرور والمحاولة مرة أخرى');
    } catch (e) {
        _showMessageDialog(
            'خطا',
            '، عفواً، حدث خطأ أثناء عملية تسجيل الدخول. يرجى المحاولة مرة أخرى');
    }
}
```

Figure 36 Login page 3

▪ Password Reset

In Nomu, we allow users to easily reset their forgotten passwords. When a user forgets their password, they can enter their email address on the dedicated page called forgot_password_page, and the app will send a password reset link to the entered email. If the email is registered in the system, the link is successfully sent, and if there's an error like an incorrect email, the user sees a message explaining the issue.

This feature was built using Firebase Authentication, and the user interface includes:

- Email Input:** A text field for the user to enter their email address.
- Email Validation:** The app checks if the email is not empty, and if there's an error with the email entered, an error message is displayed.
- Sending the Link:** The app sends a password reset link via `sendPasswordResetEmail` using Firebase.

```
try {
    await _auth.sendPasswordResetEmail(email: email);
    _showMessageDialog(
        'نجاح',
        '، إذا كان هناك حساب مسجل بهذا البريد الإلكتروني، فقد تم إرسال رابط لإعادة تعيين كلمة المرور',
        success: true, // Display the login button
    );
    setState(() {
        _linkSent = true;
    });
} on FirebaseAuthException catch (e) {
    if (e.code == 'invalid-email') {
        setState(() {
            _errorMessage = 'يرجى إدخال بريد إلكتروني صحيح';
        });
    }
}
```

Figure 37 forgot_password_page

Semester-2 1446H (Spring 2025)

4. **User Interaction:** On success, a popup message informs the user that the reset link has been sent to their email, with a button to return to the login page. If there's an error, a message is displayed explaining the cause.

▪ Logout

In the UserProfilePage, the logout process is implemented through the `_logout()` function. When the user presses the "تسجيل الخروج" button, a confirmation dialog is shown to confirm their intention to log out. This is achieved using the `showDialog` function, which presents a custom dialog with two options: "تسجيل الخروج" and "إلغاء". If the user selects "تسجيل الخروج", the logout process is triggered using `FirebaseAuth.instance.signOut()`.

1. **Confirmation Dialog:** A dialog appears to confirm the user's intention to log out.
2. **User Confirmation:** If the user chooses "تسجيل الخروج", the logout process is executed with `FirebaseAuth.instance.signOut()`.
3. **Navigation:** After logging out, the user is redirected to the login page using `Navigator.pushReplacement()`, ensuring that the current page is replaced with the login page.

```
if (shouldLogout == true) {  
    await FirebaseAuth.instance.signOut();  
    Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => LoginPage()));  
}
```

Figure 38 UserProfilePage

▪ Profile Editing

In our app, users can update their username through the `edit_profile_page` by using the `_updateProfile()` function. When the 'حفظ التعديلات' button is pressed, it first checks if the entered name is different from the current name in the app. If there is no change in the name, the process is stopped. If the name is different, a dialog is shown asking the user to confirm their desire to change the name, offering two options: 'تأكيد' to apply the change or 'إلغاء' to stop the process.

If the user agrees to change the name, the data is updated in **Firebase Authentication** and **Firestore** to ensure the new name is stored correctly.

Semester-2 1446H (Spring 2025)

```
try {
    await user.updateDisplayName(newName);

    await FirebaseFirestore.instance.collection('users').doc(user.uid).update({
        'username': newName,
    });
}
```

Figure 39 edit_profile_page

If any error occurs during the update, an error message is displayed to the user explaining the reason. If the update is successful, a confirmation message is shown, informing the user that their data has been successfully updated.

3. Lesson Module Integration

▪ Tutorial Section

In the app, a video display system has been implemented to allow users to watch educational content directly without leaving the app. The system pulls video data, including links, titles, and brief descriptions, from the Firebase Firestore database. This data is automatically retrieved when the user accesses a lesson page (LessonPage) or a playlist page (PlaylistPage) containing multiple lessons.

All educational videos shown in the app are sourced from the **Nomu Channel** on YouTube. To display these videos within the app, the **youtube_player_flutter** library is used.

```
dependencies:
  youtube_player_flutter: ^9.1.1
```

Figure 40 Tutorial Section 1

This library allows YouTube videos to be played directly within the app, without the need to open an external browser or app. The process begins by extracting the video ID from the YouTube URL using the **YoutubePlayer.convertUrlToId()** function. This ID is then passed to the **YoutubePlayerController** to initialize the player, which automatically plays the video with sound control options.

Semester-2 1446H (Spring 2025)

```
void initState() {
    super.initState();
    final videoId = YoutubePlayer.convertUrlToId(widget.videoUrl)!;
    _controller = YoutubePlayerController(
        initialVideoId: videoId,
        flags: YoutubePlayerFlags(
            autoPlay: true,
            mute: false,
        ), // YoutubePlayerFlags
    ); // YoutubePlayerController
```

Figure 41 Tutorial Section 2

This video display method enhances the user experience by reducing distractions that could result from switching between apps. It also encourages users to complete the video, as they can engage with the content without interruptions. Moreover, the automatic playback and seamless integration of video content further enhance the effectiveness of the educational process, making it easier for users to learn and focus.

- **Flash card**

The flashcard feature in the app works by displaying cards with a question on the front and an answer on the back using the `FlipCard` widget, allowing users to flip the card to review the answer. These flashcards are fetched using the `fetchFlashcardsFromFirestore` function based on the provided `lessonId`. The data for each flashcard (such as the question and answer) is extracted from documents in the `Firebase` database, with data validation (question and answer) taking place. Once validated, the flashcard is added to the internal `allCards` list.

Then, the user interface is updated to display the flashcards in the app, where users can interact with the cards by swiping right (if they understand the content) or left (if they don't and need to review it later) using the `swipe_cards` library. Flashcards that are not understood are added to a repeat list and are re-presented to the user until they are understood, which enhances the process of repetition and review.

```
swipe_cards: ^2.0.0+1
flip_card: ^0.6.0
```

Figure 43 Flash card 1

Semester-2 1446H (Spring 2025)

```
Future<void> fetchFlashcardsFromFirestore() async {
    try {
        final snapshot = await FirebaseFirestore.instance
            .collection('Learing')
            .doc(widget.lessonId)
            .collection('flashcard')
            .orderBy(FieldPath.documentId)
            .get();
    }
```

Figure 42 Flash card 1

Figure 43 Flash card 2

Once all the cards are completed and no more cards are left to review, the user is directed to a congratulations page. This mechanism combines interaction, repetition, and motivation to create an engaging and effective learning experience.

- **Daily question**

The "Daily Question" feature in the app allows users to interact with a daily question that includes multiple-choice options. Users can answer the question either from the Home Page or the Learning Page, and once answered, the question will not appear on the other page, ensuring a seamless experience. The daily question is loaded either from **SharedPreferences** or **Firebase** using the `_initializeDailyQuestion` function.

```
Future<void> _initializeDailyQuestion() async {
    _todaysQuestion = await _getDailyQuestion();
}
```

Figure 44 Daily question 1

If the user hasn't answered the question yet, it is displayed in a pop-up dialog with multiple answer options. After the user answers, the response is verified using `_showCorrectDialog` for correct answers or `_showWrongAnswerDialogWithCorrect` for incorrect answers. The answer status is updated in **Firebase** using the `_markChallengeAnswered` function, and the question is stored in **SharedPreferences** for quick access in future sessions.

Semester-2 1446H (Spring 2025)

```
Future<void> _markChallengeAnswered() async {
    final user = FirebaseAuth.instance.currentUser;
    if (user != null) {
        final now = DateTime.now();
        await FirebaseFirestore.instance.collection('users').doc(user.uid).update({
            'challenge_answered': true,
            'challenge_timestamp': Timestamp.fromDate(now),
        });
        setState(() {
            _challengeAnswered = true;
        });
    }
}
```

Figure 45 Daily question 2

If more than 24 hours have passed since the last answer, the challenge status is reset using `_loadChallengeStatus`, ensuring continued daily interaction with users.

```
Future<void> _loadChallengeStatus() async {
    final user = FirebaseAuth.instance.currentUser;
    if (user != null) {
        final userDoc = await FirebaseFirestore.instance.collection('users').doc(user.uid).get();
        if (userDoc.exists && userDoc.data() != null) {
            final challengeAnswered = userDoc['challenge_answered'] ?? false;
            final timestamp = (userDoc['challenge_timestamp'] as Timestamp)?toDate();

            if (challengeAnswered && timestamp != null) {
                final difference = DateTime.now().difference(timestamp).inHours;
                if (difference >= 24) {
                    // Reset challenge if more than 24 hours passed
                    await FirebaseFirestore.instance.collection('users').doc(user.uid).update({
                        'challenge_answered': false,
                    });
                    setState(() {
                        _challengeAnswered = false;
                    });
                } else {
                    setState(() {
                        _challengeAnswered = true;
                    });
                }
            } else {
                setState(() {
                    _challengeAnswered = false;
                });
            }
        }
    }
}
```

Figure 46 Daily question 3

4. Gamification Elements (Coin Rewards)

▪ CoinManager class

The CoinManager class encapsulates all coin-related operations, ensuring a single source of truth for in-app currency handling. The first snippet illustrates methods for retrieving the user's current balance (`getCoins`), incrementing it by a specified amount (`addCoin`), and initializing the

Semester-2 1446H (Spring 2025)

balance for new users (`initializeCoins`) . The second snippet presents two flashcard-specific routines: `hasEarnedCoinsForLesson` checks whether coins have already been granted for a particular lesson, and `markCoinsAsEarnedForLesson` records that reward in Firestore to prevent duplicate grants. This architecture guarantees precise, idempotent management of coin balances across all learning activities.

```

1 import 'package:firebase_auth/firebase_auth.dart';
2 import 'package:cloud_firestore/cloud_firestore.dart';
3
4 class CoinManager {
5     // Get the current coin balance for the logged-in user
6     static Future<int> getCoins() async {
7         final user = FirebaseAuth.instance.currentUser;
8         if (user != null) {
9             final doc = await FirebaseFirestore.instance.collection('users').doc(user.uid).get();
10            return doc.data()?['coins'] ?? 0; // Default to 0 if no coins field exists
11        }
12        return 0; // Return 0 if no user is logged in
13    }
14
15    // Add coins to the user's balance
16    static Future<void> addCoin(int amount) async {
17        final user = FirebaseAuth.instance.currentUser;
18        if (user != null) {
19            final userRef = FirebaseFirestore.instance.collection('users').doc(user.uid);
20            await userRef.update({'coins': FieldValue.increment(amount)});
21        }
22    }
23
24    // Initialize the user's coin balance if it's their first time logging in
25    static Future<void> initializeCoins() async {
26        final user = FirebaseAuth.instance.currentUser;
27        if (user != null) {
28            final userRef = FirebaseFirestore.instance.collection('users').doc(user.uid);
29            final userDoc = await userRef.get();
30            if (!userDoc.exists) {
31                await userRef.set({
32                    'coins': 0, // Start with 0 coins
33                });
34            }
35        }
36    }
37 }
```

Figure 48: coin manager

```

38     // Check if the user has earned coins for a specific flashcard lesson
39     static Future<bool> hasEarnedCoinsForLesson(String lessonId) async {
40         final user = FirebaseAuth.instance.currentUser;
41         if (user != null) {
42             final doc = await FirebaseFirestore.instance
43                 .collection('users')
44                 .doc(user.uid)
45                 .collection('earnedFlashcardCoins')
46                 .doc(lessonId)
47                 .get();
48             return doc.exists;
49         }
50         return false;
51     }
52
53     // Mark that the user has earned coins for a specific flashcard lesson
54     static Future<void> markCoinsAsEarnedForLesson(String lessonId) async {
55         final user = FirebaseAuth.instance.currentUser;
56         if (user != null) {
57             final docRef = FirebaseFirestore.instance
58                 .collection('users')
59                 .doc(user.uid)
60                 .collection('earnedFlashcardCoins')
61                 .doc(lessonId);
62             await docRef.set({'earned': true});
63         }
64     }
65 }
```

Figure 47: *coin manager*

Figure 48: *hasEarnedCoinsForLesson*

Semester-2 1446H (Spring 2025)

كلية علوم الحاسوب والمعلومات
جامعة الملك سعود

▪ LessonPage reward logic

Prior to awarding any coins, the app invokes the `checkAndRewardCoins()` method (shown below) when a lesson video finishes playing. This function first retrieves the current user's UID and locates the corresponding document in Firestore under

`users/{uid}/lessons_rewards/{lessonId}`. If no record exists, meaning the user hasn't yet been rewarded for this lesson it creates that document (storing 10 coins and a server timestamp), increments the user's overall coin balance by 10, and then displays a congratulatory dialog. This logic guarantees each lesson reward is granted exactly once.

Semester-2 1446H (Spring 2025)

```

82     Future<void> checkAndRewardCoins() async {
83         try {
84             final userId = FirebaseAuth.instance.currentUser?.uid;
85             if (userId == null || !mounted) return;
86
87             final userCoinsRef = FirebaseFirestore.instance
88                 .collection('users')
89                 .doc(userId)
90                 .collection('lessons_rewards')
91                 .doc(widget.lessonDocId);
92
93             final userCoinsDoc = await userCoinsRef.get();
94
95             if (!userCoinsDoc.exists) {
96                 await userCoinsRef.set({
97                     'coins': 10,
98                     'timestamp': FieldValue.serverTimestamp()
99                 });
100
101             final userRef = FirebaseFirestore.instance.collection('users').doc(userId);
102             await userRef.update({
103                 'coins': FieldValue.increment(10),
104             });
105
106             if (mounted) {
107                 showDialog(
108                     context: context,
109                     builder: (_) => AlertDialog(
110                         title: const Text('سيف'),
111                         content: const Text('لتحت 10 عملات إضافية التي حصلت على'),
112                         actions: [
113                             TextButton(
114                                 onPressed: () => Navigator.of(context).pop(),
115                                 child: const Text('حسناً'),
116                             ), // TextButton
117                         ],
118                     ), // AlertDialog
119                 );
120             }
121         } catch (e) {
122             debugPrint('Error rewarding coins: $e');
123         }
124     }
125 }
```

Figure 49: checkAndRewardCoins() for lessons

- **Tutorials videos reward logic**

The `checkAndRewardCoins()` method in `VideoPlayerPage.dart` ensures that users earn coins only once for watching each tutorial video. When the YouTube player reaches the ended state, this function retrieves the current user's UID and constructs a reference to `users/{uid}/lessons_rewards/{videoTitle}` in Firestore. It then checks whether a

reward document already exists; if not, it creates one with a 10-coin entry and atomically increments the user's overall coins field by 10. Finally, it displays an RTL confirmation dialog notifying the user of their newly earned coins. This approach guarantees idempotent, per-video coin awards.

```
44     Future<void> checkAndRewardCoins() async {
45       final userId = FirebaseAuth.instance.currentUser?.uid;
46       if (userId == null) return;
47
48       final userCoinsRef = FirebaseFirestore.instance
49           .collection('users')
50           .doc(userId)
51           .collection('lessons_rewards')
52           .doc(widget.videoTitle);
53
54       final userCoinsDoc = await userCoinsRef.get();
55
56       if (!userCoinsDoc.exists) {
57         await userCoinsRef.set({'coins': 10});
58
59         final userRef = FirebaseFirestore.instance.collection('users').doc(userId);
60         await userRef.update({
61           'coins': FieldValue.increment(10),
62         });
63
64         if (mounted) {
65           showDialog(
66             context: context,
67             builder: (_) => AlertDialog(
68               title: Text('مبروك'),
69               content: Text('تحت 10 عملات لفافيه الى صبيك'),
70               actions: [
71                 TextButton(
72                   onPressed: () {
73                     Navigator.of(context).pop();
74                   },
75                   child: Text('حسناً'),
```

Figure 50: *checkAndRewardCoins()* for tutorials

▪ Flashcard Completion Rewards

When a user finishes all cards in a lesson, the congratulations dialog's confirmation button executes this logic. It first checks via `hasEarnedCoinsForLesson` whether the 10-coin reward for that lesson has already been granted. If not, it awards 10 coins by calling `addCoin(10)` and then records the reward in Firestore with `markCoinsAsEarnedForLesson`, ensuring the bonus cannot be re-claimed. Finally, it closes the dialog and returns true so the parent screen can refresh the displayed balance.

```
FutureBuilder<bool>(
    future: CoinManager.hasEarnedCoinsForLesson(lessonId),
    builder: (context, snapshot) {
        if (snapshot.connectionState != ConnectionState.done) {
            return CircularProgressIndicator(color: Colors.green);
        }

        final hasEarned = snapshot.data ?? false;
        final message = hasEarned
            ? "فينا! أنت الآن حاصل على جائزة الامتحان"
            : "تهانينا! تمت إضافة 10 عملات إلى محفظتك \n كل خطوة تعلم تقربك نحو النجاح!";

        return Text(
            message,
            textAlign: TextAlign.center,
            style: TextStyle(
                fontSize: 22,
                color: Colors.black87,
                fontWeight: FontWeight.bold,
            ), // TextStyle
        ); // Text
    },
), // FutureBuilder
SizedBox(height: 40),
ElevatedButton(
    onPressed: () async {
        bool hasEarned = await CoinManager.hasEarnedCoinsForLesson(lessonId);
        if (!hasEarned) {
            await CoinManager.addCoin(10);
            await CoinManager.markCoinsAsEarnedForLesson(lessonId);
        }
        Navigator.pop(context, true);
    },
    child: Text("إعوامدة النجاح", style: TextStyle(fontSize: 16)),
);
```

Figure 51 :`hasEarnedCoinsForLesson()`

Semester-2 1446H (Spring 2025)

Daily Challenge Rewards

Upon correctly answering the daily quiz, the “موافق” button handler runs this code. It retrieves the current user's document in Firestore and atomically increments their coins field by 5. After updating the balance, it navigates back to the Home screen and marks the challenge as answered for the day, preventing further coin awards until the next 24-hour cycle.

```
void _showCorrectDialog() {
    showDialog(
        context: context,
        barrierDismissible: false,
        builder: (ctx) => Stack(children: [
            SizedBox.expand(child: Lottie.asset('assets/fireworks.json', fit: BoxFit.cover, repeat: true)),
            Center(
                child: Column(mainAxisSize: MainAxisSize.min, children: [
                    Text("الإجابة صحيحة", style: TextStyle(fontSize: 40, color: Colors.white, fontWeight: FontWeight.bold)),
                    SizedBox(height: 30),
                    ElevatedButton(
                        onPressed: () async {
                            final user = FirebaseAuth.instance.currentUser;
                            if (user != null) {
                                await FirebaseFirestore.instance.collection('users').doc(user.uid)
                                    .update({'coins': FieldValue.increment(5)});
                            }
                            Navigator.pushReplacement(context, MaterialPageRoute(builder: (_)> HomePage()));
                            _markChallengeAnswered();
                        },
                        style: ElevatedButton.styleFrom(backgroundColor: Color(0xFF90C08B)),
                        child: Text("موافق"),
                    ) // ElevatedButton
    ])));
}
```

Figure 52: Daily Challenge Rewards

5. Market Data Integration and Simulation Engine

▪ Market Overview Chart Logic

The Market Overview Chart provides interactive multi-company visualization that helps users gain insight into broad market movements. The bottom-sheet filtering interface allows users to select up to five companies using dynamically generated checkbox tiles. When the user confirms their selection, the page resolves the corresponding company IDs, preloads missing historical data through `_loadCompanyDataToCache()`, and rebuilds the entire chart by calling `_buildSeriesFromCache()`. This ensures efficient updates and accurate plotting at all times.

To support flexible market analysis, the chart integrates custom `_TfChip` widgets that switch between different time windows such as 1W, 1M, and 3M. These chips animate selections and apply Nomu's green color theme for consistency across the simulation module.

Together, these components provide a highly responsive and intuitive charting experience tailored to Arabic RTL users.

Figure 54 shows the filtering logic, Figure 55 displays the chip implementation, and Figure 56 illustrates the chart-rebuild workflow.

Semester-2 1446H (Spring 2025)

```
ElevatedButton(  
    onPressed: () async {  
        Navigator.pop(context);  
  
        // Find IDs for selected names and Load Data if missing  
        final selectedIds = <String>[];  
        _companyMeta.forEach((id, meta) {  
            if (tmpSelected.contains(meta['name'])) selectedIds.add(id);  
        });  
  
        // Show loading indicator in parent if needed, or just await  
        await Future.wait(selectedIds.map((id) => _loadCompanyDataToCache(id)));  
  
        if (mounted) {  
            setState(() {  
                _selectedCompanies..clear()..addAll(tmpSelected);  
                _future = Future.value(_buildSeriesFromCache(_window)); // Rebuild  
                _zoomPanBehavior.reset();  
            });  
        }  
    },  
    child: const Text('الغاء'),  
,
```

Figure 54:filtering logic

Semester-2 1446H (Spring 2025)

```
class _TfChip extends StatelessWidget {
    final String label;
    final bool selected;
    final VoidCallback onTap;
    const _TfChip({required this.label, required this.selected, required this.onTap});

    @override
    Widget build(BuildContext context) {
        const color = Color(0xFF609966);

        return GestureDetector(
            onTap: onTap,
            child: AnimatedContainer(
                duration: const Duration(milliseconds: 150),
                padding: const EdgeInsets.symmetric(horizontal: 12, vertical: 6),
                decoration: BoxDecoration(
                    color: selected ? color.withOpacity(0.12) : Colors.grey.shade200,
                    borderRadius: BorderRadius.circular(999),
                    border: Border.all(
                        color: selected ? color : Colors.grey.shade300,
                        width: selected ? 1.6 : 1),
                ),
                child: Text(label,
                    style: TextStyle(
                        fontWeight: FontWeight.w600,
                        color: selected ? color : Colors.black87)),
            ),
        );
    }
}
```

Figure 55:the chip implementation

```
Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
        _TfChip(label: '1W', selected: _window == MarketWindow.w1, onTap: () => _setWindow(MarketWindow.w1)),
        const SizedBox(width: 8),
        _TfChip(label: '1M', selected: _window == MarketWindow.m1, onTap: () => _setWindow(MarketWindow.m1)),
        const SizedBox(width: 8),
        _TfChip(label: '3M', selected: _window == MarketWindow.m3, onTap: () => _setWindow(MarketWindow.m3)),
    ],
),
```

Figure 56:the chart-rebuild workflow

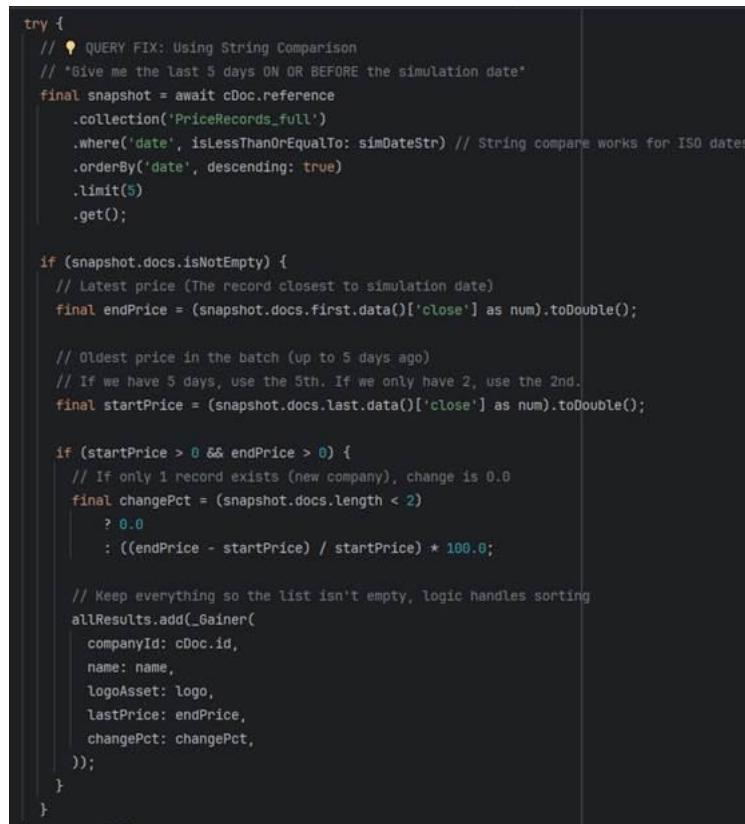
Semester-2 1446H (Spring 2025)

■ Top Five Weekly Performers Logic

The simulation module highlights the strongest market movers through a “Top Five Weekly Performers” section. This feature computes weekly gains by comparing each company’s most recent price record with its value at the beginning of the simulated week. Using historical data stored in Firestore, the algorithm calculates percentage change for every company, sorts them in descending order, and selects the top five performers.

Each company is presented in a stylized RTL card containing its logo, name, current price, and weekly percentage gain. Tapping any entry navigates directly to its detailed chart in the CompanyChartWidget, creating a seamless transition between market ranking and in-depth analysis. This feature helps learners quickly identify high-performing stocks and explore potential patterns inside the simulation environment.

Figures 57-59 shows the weekly-gainer calculation.



```
try {
    // QUERY FIX: Using String Comparison
    // *Give me the last 5 days ON OR BEFORE the simulation date*
    final snapshot = await cDoc.reference
        .collection("PriceRecords_full")
        .where('date', isLessThanOrEqual: simDateStr) // String compare works for ISO dates
        .orderBy('date', descending: true)
        .limit(5)
        .get();

    if (snapshot.docs.isNotEmpty) {
        // Latest price (The record closest to simulation date)
        final endPrice = (snapshot.docs.first.data()['close'] as num).toDouble();

        // Oldest price in the batch (up to 5 days ago)
        // If we have 5 days, use the 5th. If we only have 2, use the 2nd.
        final startPrice = (snapshot.docs.last.data()['close'] as num).toDouble();

        if (startPrice > 0 && endPrice > 0) {
            // If only 1 record exists (new company), change is 0.0
            final changePct = (snapshot.docs.length < 2)
                ? 0.0
                : ((endPrice - startPrice) / startPrice) * 100.0;

            // Keep everything so the list isn't empty, logic handles sorting
            allResults.add(_Gainer(
                companyId: cDoc.id,
                name: name,
                logoAsset: logo,
                lastPrice: endPrice,
                changePct: changePct,
            ));
        }
    }
}
```

Figure 58:weekly-gainer calculation2

Semester-2 1446H (Spring 2025)

```
        } catch (_) {}
    });

    processedCount += batch.length;
    if (mounted) {
        setState(() {
            _loadingPercent = processedCount / totalCompanies;
            _loadingProgressText = "${(processedCount / totalCompanies * 100).toInt()}%";
        });
    }
    await Future.delayed(const Duration(milliseconds: 1));
}

// Sort: Highest gains first
allResults.sort((a, b) => b.changePct.compareTo(a.changePct));

if (mounted) {
    setState(() {
        // Take Top 5. If list is empty, UI will show "No Data" gracefully.
        _cachedWeeklyGainers = allResults.take(5).toList();
        _isLoadingGainers = false;
    });
}
} catch (e) {
    debugPrint("Error loading gainers: $e");
    if (mounted) setState(() => _isLoadingGainers = false);
} finally {
    _msgTimer?.cancel();
}
}
```

Figure 59:weekly-gainer calculation3

6. CompanyChart & Trading Logic

When a user selects a company from the simulation page, the system loads its historical price records and plots them into an RTL chart optimized for Arabic learners. The `_convertToRealDate()` method aligns simulated trading days with real Tadawul market days by skipping Fridays and Saturdays, ensuring that all plotted data reflects authentic Saudi trading behavior.

Users can initiate a transaction directly from this screen. When the Buy or Sell button is pressed, the widget displays a localized Arabic confirmation sheet that summarizes the order using `_buildReceiptRow()`, which formats details such as the stock symbol, selected quantity, unit price, and total value. Confirming the trade triggers the `TradingService.placeTrade()` function, which records the transaction under the user's Firestore orders collection, updates

Semester-2 1446H (Spring 2025)

open positions, and recalculates wallet balances accordingly. This workflow guarantees that every buy or sell action immediately impacts the user's simulated portfolio and wallet.

Figure 60 illustrates the date-conversion logic, Figure 61 shows the Arabic receipt layout, and Figure 62 displays the buy/sell transaction implementation.

```
DateTime _convertToRealDate(  
    DateTime firebaseDate, List<DateTime> allTradingDays, DateTime simDate) {  
    int indexInSim = allTradingDays.indexWhere((d) =>  
        d.year == firebaseDate.year &&  
        d.month == firebaseDate.month &&  
        d.day == firebaseDate.day);  
    if (indexInSim == -1) return firebaseDate;  
  
    int currentIndex = allTradingDays.indexWhere((d) =>  
        d.year == simDate.year &&  
        d.month == simDate.month &&  
        d.day == simDate.day);  
    if (currentIndex == -1) return firebaseDate;  
  
    int daysDiff = currentIndex - indexInSim;  
  
    DateTime realToday = DateTime.now();  
    if (realToday.weekday == DateTime.friday) {  
        realToday = realToday.subtract(const Duration(days: 1));  
    } else if (realToday.weekday == DateTime.saturday) {  
        realToday = realToday.subtract(const Duration(days: 2));  
    }  
  
    DateTime resultDate = realToday;  
    int tradingDaysSubtracted = 0;  
    while (tradingDaysSubtracted < daysDiff) {  
        resultDate = resultDate.subtract(const Duration(days: 1));  
        if (resultDate.weekday != DateTime.friday &&  
            resultDate.weekday != DateTime.saturday) {  
            tradingDaysSubtracted++;  
        }  
    }  
    return resultDate;  
}
```

Figure 60:the date-conversion logic

Semester-2 1446H (Spring 2025)

```
Widget _buildReceiptRow(String label, String value, {Color? valueColor, bool isBold = false}) {
    return Padding(
        padding: const EdgeInsets.only(bottom: 8),
        child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
                Text(label, style: TextStyle(color: Colors.grey.shade600, fontSize: 13)),
                Text(
                    value,
                    style: TextStyle(
                        color: valueColor ?? Colors.black87,
                        fontWeight: isBold ? FontWeight.bold : FontWeight.w500,
                        fontSize: 14
                    ),
                ),
            ],
        ),
    );
}
```

Figure 61:the Arabic receipt layout

Semester-2 1446H (Spring 2025)

```

final cost = price * qty;

if (side == 'BUY') {
    if (cash < cost) {
        throw Exception('INSUFFICIENT_CASH');
    }
    // avg cost
    final newQty = currentQty + qty;
    final newAvg = (currentQty == 0) ? price : ((avgCost * currentQty) + cost) / newQty;

    // ensure normalized ref
    posRef = await _upsertPositionRef(uid, symbol: symbol, companyName: name);

    tx.set(
        posRef,
        {
            'symbol': symbol,
            'name': name,
            'qty': newQty,
            'avgCost': double.parse(newAvg.toStringAsFixed(6)),
            'updatedAt': FieldValue.serverTimestamp(),
        },
        SetOptions(merge: true),
    );

    tx.update(walletRef, {'cash': cash - cost});
} else {
    // SELL
    if (currentQty < qty) {
        throw Exception('INSUFFICIENT SHARES');
    }

    // ensure normalized ref
    posRef = await _upsertPositionRef(uid, symbol: symbol, companyName: name);

    final remain = currentQty - qty;
    if (remain == 0) {
        tx.delete(posRef);
    } else {
        tx.set(
            posRef,
            {
                'qty': remain,
                // keep same avgCost
                'avgCost': avgCost,
                'symbol': symbol,
                'name': name,
                'updatedAt': FieldValue.serverTimestamp(),
            },
            SetOptions(merge: true),
        );
    }

    tx.update(walletRef, {'cash': cash + cost});
}

```

Figure 62:the buy/sell transaction implementation.

7. Market Prediction Module

The Prediction Model feature displays AI-generated forecasts for each company based on pre-computed values stored in Firestore. When the page loads, the `_bootstrapPage()` method retrieves the current simulation date, loads the user's favorites, and fetches both actual historical prices and predicted values. During this process, the interface shows rotating status messages from `_loadingMessages` to guide the user through each loading stage.

For every company, the model compares the latest closing price with the predicted value and calculates the percentage difference. Based on this comparison, the system generates a simple recommendation Buy if the predicted value is higher, or Don't Buy otherwise. All results are presented in an RTL-friendly list, allowing users to quickly understand model-based signals without interacting with complex machine-learning details.

Figure 63 shows the bootstrap logic, and Figure 64 illustrates the prediction comparison and recommendation rule.

```
Future<void> _bootstrapPage() async {
    // Stage 1: Load User & Favorites
    setState(() => _loadingProgressText = "20%");
    await _loadFavorites();

    // Stage 2: Calculate Simulation Date
    setState(() => _loadingProgressText = "45%");
    await _loadSimulationAnchorAndCompute();

    // Stage 3: Load Companies
    setState(() => _loadingProgressText = "70%");
    await _loadInitialCompanies();

    // Finish
    setState(() => _loadingProgressText = "100%");
}
```

Figure 63:the bootstrap logic

```
// Re-fetch prediction if we switched to a fallback date
if (!predDoc.exists && _cachedLatestAvailableDateId != null) {
    predDoc = await _withTimeout(predCol.doc(_cachedLatestAvailableDateId).get());
}

if (!priceDoc.exists || !predDoc.exists) {
    return {'insufficient': true, 'message': 'بيانات غير متوفرة'};
}

final latestClose = (priceDoc.data()?[['close']] as num?)?.toDouble() ?? 0.0;
final predicted = (predDoc.data()?[['predicted']] as num?)?.toDouble();

if (predicted == null || latestClose == 0) {
    return {'insufficient': true, 'message': 'بيانات غير متوفرة'};
}

final diff = predicted - latestClose;
final pct = (diff / latestClose) * 100.0;
final decisionAr = (predicted < latestClose) ? 'بيع' : 'شراء';

return {
    'close': latestClose,
    'predicted': predicted,
    'decisionAr': decisionAr,
    'pct': pct,
    'simDate': _currentSimDate!.toIso8601String(),
};
```

Figure 64:the prediction comparison and recommendation rule.

8. Portfolio Page (Performance Tracking)

The PortfolioPage reconstructs the user's simulated investment journey.

The `_loadChart()` method generates historical portfolio values by replaying past orders day by day, storing them in `_chartDays` and `_chartTotals` arrays.

`_calcPnLFromOrders()` iterates through all transactions, revaluing each open lot at historical closing prices and aggregating total profit, loss, and cost.

The final data feed is rendered via the `f1_chart` library as an animated RTL line chart showing performance over the selected period.

Figures 65–67 depict the chart computation and P&L calculation logic.

Semester-2 1446H (Spring 2025)

```

if (_range == '1D') {
    // ====== أداء اليوم (baseline + realized from today's sells) ======
    // التكلفة المتداولة قبل اليوم وما قبل اليوم
    final todayOrders = <QueryDocumentSnapshot<Map<String, dynamic>>>[];
    final beforeTodayOrders = <QueryDocumentSnapshot<Map<String, dynamic>>>[];
    for (final d in orders) {...}
    todayOrders.sort((a, b) => _extractDate(a.data()).compareTo(_extractDate(b.data())));

    // قيم العروض المتداولة في اليوم (بيانات البيع)
    final startRes = await _calcPnlFromOrders(beforeTodayOrders, ignoreSameDay: true);
    final currentValueStart = startRes.currentValue;

    // تكاليف متداولة في اليوم = التكاليف الحالى - تكاليف متداولة
    double netCashChangeToday = 0.0;
    for (final d in todayOrders) {...}
    final cashAtStart = cash - netCashChangeToday;

    // تحويل من ما قبل اليوم
    final fifoBySymbol = <String, List<Map<String, double>>>{};
    for (final lot in _buildOpenLotsFromOrders(beforeTodayOrders)) {...}

    final baselineTotal = cashAtStart + currentValueStart;

    final totals = <double>[baselineTotal];
    final days = <DateTime>[DateTime(todayReal.year, todayReal.month, todayReal.day, 9, 30)];

    double realizedToday = 0.0;

    for (final d in todayOrders) {...}

    // --- تأكيد أن آخر بيع == احتمال المحفظة الحالى (مثل الكرونة) ---
    final totalNow = await _resolveTotalPortfolioNow(orders, cash);
    if (totals.isEmpty) {...} else {...}

    _chartTotals = totals;
    _chartDays = days;
} else {
    // 1W/1M/3M
}

```

Figure 65:the chart computation and P&L calculation logic 1

Semester-2 1446H (Spring 2025)

كلية علوم الحاسوب والمعلومات
قسم تكنولوجيا المعلومات

```
    } else {
        // 1W/1M/3M
        final count = _range == '1W' ? 7 : (_range == '1M' ? 30 : 90);

        // اثنى عشر يوماً
        final days = <DateTime>[];
        DateTime cur = todayReal;
        while (days.length < count) {
            if (SimulationUtils.isWorkday(cur)) days.add(cur);
            cur = cur.subtract(const Duration(days: 1));
        }
        days.sort();

        // اثنى عشر
        final totals = <double>[];
        for (final d in days) {
            final v = await _portfolioAtDay(d, orders, cash, firstOrderReal);
            totals.add(v);
        }
        _chartDays = days;
        _chartTotals = totals;

        // ----- تأكيد أن آخر نقطة الممتدة العائدة (ستة نقاط) == -----  

        if (_chartTotals.isNotEmpty) {
            _chartTotals[_chartTotals.length - 1] = await _resolveTotalPortfolioNow(orders, cash);
        }
    }

    _chartLoading = false;
    setState(() {});
}
```

Figure 66:the chart computation and P&L calculation logic 2

Semester-2 1446H (Spring 2025)

```

Future<{double totalProfit, double totalLoss, double totalCost, double currentValue}> _calcPnLFromOrders(
    List<QueryDocumentSnapshot<Map<String, dynamic>>> orderDocs,
    bool ignoreSameDay = true,
) async {
    final simDatasetDay = await _effectivePricingDayNow();
    final openLots = _buildOpenLotsFromOrders(orderDocs);

    double totalProfit = 0.0;
    double totalLoss = 0.0;
    double totalCost = 0.0;
    double currentValue = 0.0;

    final Map<String, double> closeCache = {};

    for (final lot in openLots) {
        if (ignoreSameDay) {
            final lotDatasetDay = await _effectivePricingDayForBuy(lot.buyDate);
            if (_sameYMD(lotDatasetDay, simDatasetDay)) {
                continue; // skip same P&L
            }
        }

        final symbol = lot.symbol;
        final close = closeCache.containsKey(symbol)
            ? closeCache[symbol]!
            : (await _latestCloseUntil(symbol, simDatasetDay) ?? 0.0);
        closeCache[symbol] = close;

        final nowVal = close * lot.qty;
        final cost = lot.price * lot.qty;
        final diff = nowVal - cost;

        totalCost += cost;
        currentValue += nowVal;

        if (diff > 0) totalProfit += diff;
        else if (diff < 0) totalLoss += -diff;
    }

    return (totalProfit: totalProfit, totalLoss: totalLoss, totalCost: totalCost, currentValue: currentValue);
}
    
```

Figure 67:the chart computation and P&L calculation logic 3

9. Wallet Management

The WalletPage centralizes balance tracking and provides a controlled reset feature for testing.

The `_resetWallet()` function displays a confirmation dialog (Figure 68) before restoring the virtual balance to 10 000 SAR.

Once confirmed, it rewrites the Firestore wallet documents (`users/{uid}/wallet/main` and `Main`) and purges the orders and positions subcollections to reinitialize the simulation.

Immediate feedback is shown through an RTL Snackbar notification.

This feature safeguards users from accidental data loss while supporting repeat training cycles.

Figures 68–69 highlight the confirmation prompt and reset execution logic.

Semester-2 1446H (Spring 2025)

```
Future<void> _resetWallet() async {
    final ok = await showDialog<bool>(
        context: context,
        builder: (ctx) => AlertDialog(
            backgroundColor: const Color(0xFFFFE5E5),
            shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(16)),
            title: const Text('تحذير', textAlign: TextAlign.right),
            content: const Text(
                'أنت متى حفظت محفظة على جهازك؟',
                style: TextStyle(fontSize: 16, color: Colors.black),
            ),
            actions: [
                TextButton(
                    style: TextButton.styleFrom(
                        backgroundColor: const Color(0xFFE5E7EB),
                        foregroundColor: Colors.black,
                    ),
                    onPressed: () => Navigator.pop(ctx, false),
                    child: const Text('إلغاء'),
                ),
                TextButton(
                    style: TextButton.styleFrom(
                        backgroundColor: Colors.red,
                        foregroundColor: Colors.white,
                        shape: RoundedRectangleBorder(
                            borderRadius: BorderRadius.circular(8),
                        ),
                    ),
                    onPressed: () => Navigator.pop(ctx, true),
                    child: const Text('تأكيد'),
                ),
            ],
        );
    );
    if (ok != true) return;
}
```

Figure 68:the confirmation prompt

```
final uid = FirebaseAuth.instance.currentUser?.uid;
if (uid == null) return;

try {
    final walletCol = FirebaseFirestore.instance
        .collection('users')
        .doc(uid)
        .collection('wallet');

    // main/Main في كل من الكاتب إلى 10000
    await walletCol.doc('main').set({'cash': _seed}, SetOptions(merge: true));
    await walletCol.doc('Main').set({'cash': _seed}, SetOptions(merge: true));

    // 10,000
    await _deleteUserSubcollection(uid, 'orders');
    await _deleteUserSubcollection(uid, 'positions');

    if (!mounted) return;
    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('تم إضافة 10,000 درهم إلى محفظتك')),);
} catch (e) {
    if (!mounted) return;
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('حدث خطأ: $e')));
}
```

Figure 69:reset execution logic

10. Notification

- Notification Service

The `NotificationService` class integrates Firebase Cloud Messaging with the `flutter_local_notifications` package.

It creates daily reminders through `zonedSchedule()` while managing FCM token registration in Firestore for remote notifications.

Each reminder is bound to a unique Android channel to ensure compatibility across OS versions.

The notification logic is fully localized in Arabic and respects user preferences stored in `SharedPreferences`.

Figure 70-71 shows the scheduling workflow.

Semester-2 1446H (Spring 2025)

```

Future<void> _scheduleDaily(int hour, int minute, int id, String title, String body) async {
    await _localNotifications.zonedSchedule(
        id,
        title,
        body,
        _nextInstanceOfTime(hour, minute),
        const NotificationDetails(
            android: AndroidNotificationDetails(
                'scheduled_channel',
                'Scheduled Reminders',
                importance: Importance.defaultImportance,
                priority: Priority.high,
            ),
            androidScheduleMode: AndroidScheduleMode.exactAllowWhileIdle,
            uiLocalNotificationDateInterpretation: UILocalNotificationDateInterpretation.absoluteTime,
            matchDateTimeComponents: DateTimeComponents.time,
        );
}

tz.TZDateTime _nextInstanceOfTime(int hour, int minute) {
    final tz.TZDateTime now = tz.TZDateTime.now(tz.local);
    tz.TZDateTime scheduledDate = tz.TZDateTime(tz.local, now.year, now.month, now.day, hour, minute);
    if (scheduledDate.isBefore(now)) {
        scheduledDate = scheduledDate.add(const Duration(days: 1));
    }
    return scheduledDate;
}

Future<void> _saveNotificationToHistory(String? title, String? body) async {
    User? user = _auth.currentUser;
    if (user != null && title != null) {
        await _firestore.collection('users').doc(user.uid).collection('notifications').add({
            'title': title,
            'body': body,
            'timestamp': FieldValue.serverTimestamp(),
            'read': false,
        });
    }
}

```

Figure 70:the scheduling workflow1

```

// --- HELPERS ---

Future<void> saveTokenToDatabase([String? token]) async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    if ((prefs.getBool('notifications_enabled') ?? true) == false) return;

    User? user = _auth.currentUser;
    if (user != null) {
        String? currentToken = token ?? await _fcm.getToken();
        if (currentToken != null) {
            await _firestore.collection('users').doc(user.uid).set({
                'fcmToken': currentToken,
                'lastActive': FieldValue.serverTimestamp(),
            }, SetOptions(merge: true));
        }
    }
}

```

Figure 71:the scheduling workflow2

Semester-2 1446H (Spring 2025)

- Notification Settings and History

The `NotificationSettingsPage` provides users with granular control over reminders and push messages.

A switch widget toggles in-app alerts by calling

```
NotificationService.enableNotifications() or
disableNotifications(), persisting the state via SharedPreferences.
```

The `NotificationsPage` retrieves all notifications from `users/{uid}/notifications` and lists them in RTL cards with titles, messages, and timestamps.

A single-action “clear all” button performs a Firestore batch delete for efficient cleanup.

Figures 72–75 demonstrate the settings toggle and notification-list rendering.

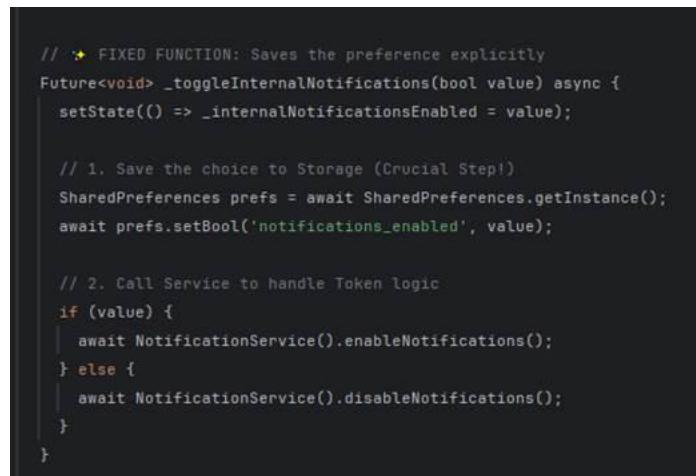
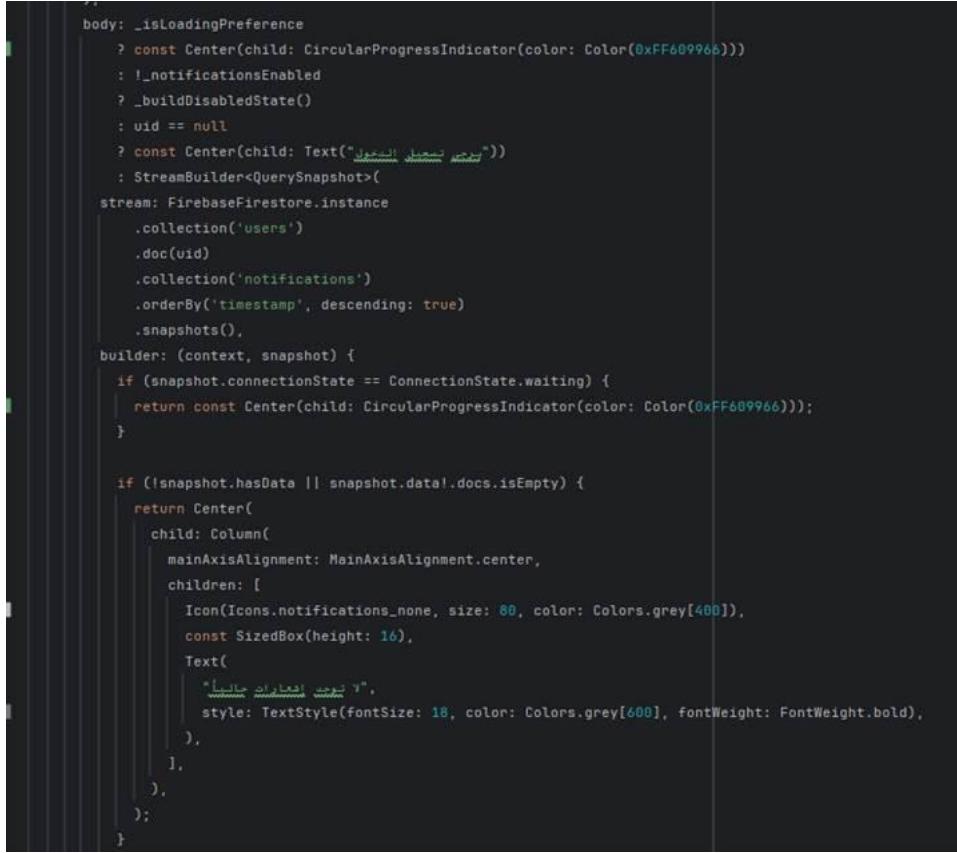


Figure 72:Toggle handler logic



Figure 73:SwitchListTile UI rendering

Semester-2 1446H (Spring 2025)



```
body: _isLoadingPreference
    ? const Center(child: CircularProgressIndicator(color: Color(0xFF609966)))
    : !_notificationsEnabled
    ? _buildDisabledState()
    : uid == null
    ? const Center(child: Text("لا تمتلك إشعارات"))
    : StreamBuilder<QuerySnapshot>(
        stream: FirebaseFirestore.instance
            .collection('users')
            .doc(uid)
            .collection('notifications')
            .orderBy('timestamp', descending: true)
            .snapshots(),
        builder: (context, snapshot) {
            if (snapshot.connectionState == ConnectionState.waiting) {
                return const Center(child: CircularProgressIndicator(color: Color(0xFF609966)));
            }

            if (!snapshot.hasData || snapshot.data!.docs.isEmpty) {
                return Center(
                    child: Column(
                        mainAxisAlignment: MainAxisAlignment.center,
                        children: [
                            Icon(Icons.notifications_none, size: 80, color: Colors.grey[400]),
                            const SizedBox(height: 16),
                            Text(
                                "لا تمتلك إشعارات",
                                style: TextStyle(fontSize: 18, color: Colors.grey[600], fontWeight: FontWeight.bold),
                            ),
                        ],
                );
            }
        });
    
```

Figure 74:Notification list structure and loading state

11. Favorites List Functionality

▪ Fetching Favorites

The `fetchEducationFavorites()` method retrieves all of a user's favorite lessons and videos in one go. It first obtains the current user's ID from FirebaseAuth (returning early if null), then initializes an empty list. It queries Firestore at `Favorites/{userId}/lessons` to get every favorite-lesson document, extracting each lesson's ID and stored title. For each lesson, it also reads the original Lessons collection to see if it has a `playlist_videos` array—if not (or it's empty), the code treats it as a single-video lesson and adds an entry of type “lesson” (with id and title) to the list. It then fetches the videos subcollection under each `Favorites/{userId}/lessons/{lessonId}`, and for every video document adds an entry of type “video” with its id, subtitle (as title), URL, and parent `lessonId`. Finally, it calls `setState()` to update the UI with this combined list of favorite lessons and videos.

```

    Future<void> fetchEducationFavorites() async {
        final userId = FirebaseAuth.instance.currentUser?.uid;
        if (userId == null) return;

        List<Map<String, dynamic>> items = [];

        final lessonsSnap = await FirebaseFirestore.instance
            .collection('Favorites')
            .doc(userId)
            .collection('lessons')
            .get();

        for (var doc in lessonsSnap.docs) {
            final lessonId = doc.id;
            final title = doc['title'];

            final lessonSnapshot = await FirebaseFirestore.instance
                .collection('Learning')
                .doc(lessonId)
                .get();

            final lessonData = lessonSnapshot.data();

            final isSingleVideo = lessonData == null ||
                !lessonData.containsKey('playlist_videos') ||
                (lessonData['playlist_videos'] as List).isEmpty;

            if (isSingleVideo) {
                items.add({'id': lessonId, 'title': title, 'type': 'lesson'});
            }

            final videosSnap = await FirebaseFirestore.instance
                .collection('Favorites')
                .doc(userId)
                .collection('lessons')
                .doc(lessonId)
                .collection('videos')
                .get();

            for (var videoDoc in videosSnap.docs) {
                items.add({
                    'id': videoDoc.id,
                    'title': videoDoc['subtitle'],
                    'url': videoDoc['url'],
                    'lessonId': lessonId,
                });
            }
        }
    }
}

```

Figure 53: `fetchEducationFavorites()`

■ Removing Favorites

The `removeFavorite(item)` method handles deleting a favorite when a user taps “unfavorite.” It takes an item map (containing at least id and type, and lessonId for videos), retrieves the userId, and returns early if null. It then switches on item['type']:

- For "lesson", it deletes `Favorites/{userId}/lessons/{id}`.
- For "video", it deletes `Favorites/{userId}/lessons/{lessonId}/videos/{id}`.
- For "flashcard", it deletes `Favorites/{userId}/flashcards/{id}`.

Semester-2 1446H (Spring 2025)

After the Firestore deletion completes, it calls `setState()` to remove the matching element from the local `educationFavorites` list (matching by id and type, and for videos also by `lessonId`), ensuring the UI immediately reflects the change.

```
Future<void> removeFavorite(Map<String, dynamic> item) async {
    final userId = FirebaseAuth.instance.currentUser?.uid;
    if (userId == null) return;

    if (item['type'] == 'lesson') {
        await FirebaseFirestore.instance
            .collection('Favorites')
            .doc(userId)
            .collection('lessons')
            .doc(item['id'])
            .delete();
    } else if (item['type'] == 'video') {
        await FirebaseFirestore.instance
            .collection('Favorites')
            .doc(userId)
            .collection('lessons')
            .doc(item['lessonId'])
            .collection('videos')
            .doc(item['id'])
            .delete();
    } else if (item['type'] == 'flashcard') {
        await FirebaseFirestore.instance
            .collection('Favorites')
            .doc(userId)
            .collection('flashcards')
            .doc(item['id'])
            .delete();
    }

    setState(() {
        educationFavorites.removeWhere((element) =>
            element['id'] == item['id'] &&
            element['type'] == item['type'] &&
            (item['type'] != 'video' || element['lessonId'] == item['lessonId']));
    });
}
```

Figure 54: Removing Favorites

▪ Toggling Lesson Favorites

The favorite icon feature in the app works by displaying a heart icon through the FavoriteLessonIcon widget, which takes a lessonId and retrieves the current user's ID via FirebaseAuth. When the widget is first inserted into the UI, it calls the asynchronous `checkIfFavorite()` method. This method constructs a Firestore reference to `Favorites/{userId}/lessons/{lessonId}` and attempts to read that document. If the document exists, it sets the `isFavorite` flag to true inside a `setState()` call so the icon appears filled. Any errors during this read are caught and logged with `debugPrint()`, ensuring the interface remains responsive even if the lookup fails.

```
class FavoriteLessonIcon extends StatefulWidget {
    final String lessonId;

    const FavoriteLessonIcon({required this.lessonId});

    @override
    _FavoriteLessonIconState createState() => _FavoriteLessonIconState();
}

class _FavoriteLessonIconState extends State<FavoriteLessonIcon> {
    bool isFavorite = false;
    final userId = FirebaseAuth.instance.currentUser?.uid;

    @override
    void initState() {
        super.initState();
        checkIfFavorite();
    }

    Future<void> checkIfFavorite() async {
        if (userId == null) return;

        try {
            final doc = await FirebaseFirestore.instance
                .collection('Favorites')
                .doc(userId)
                .collection('lessons')
                .doc(widget.lessonId)
                .get();

            if (mounted) {
                setState(() {
                    isFavorite = doc.exists;
                });
            }
        } catch (e) {
            debugPrint('Error checking favorite: $e');
        }
    }

    Future<void> toggleFavorite() async {
        if (userId == null) return;

        try {
            final favRef = FirebaseFirestore.instance
                .collection('Favorites')
                .doc(userId)
                .collection('lessons')
                .doc(widget.lessonId);
        } catch (e) {
            debugPrint('Error toggling favorite: $e');
        }
    }
}
```

Figure 55: Toggling Lesson Favorites

Semester-2 1446H (Spring 2025)

Tapping the heart icon invokes the asynchronous `toggleFavorite()` method, which first guards against a null `userId` before building the same Firestore document reference. If `isFavorite` is true, it deletes the document; otherwise, it fetches the lesson's title from the `Learing` collection and writes a new document containing `lessonId`, `title`, and a server-generated timestamp via `FieldValue.serverTimestamp()`. Once the Firestore operation completes, it flips `isFavorite` inside `setState()` to immediately refresh the icon. Any exceptions during these writes or deletes are caught and printed with `debugPrint()`, preserving a smooth user experience.

```
Future<void> toggleFavorite() async {
    if (userId == null) return;

    try {
        final favRef = FirebaseFirestore.instance
            .collection('Favorites')
            .doc(userId)
            .collection('lessons')
            .doc(widget.lessonId);

        if (isFavorite) {
            await favRef.delete();
        } else {
            final lessonDoc = await FirebaseFirestore.instance
                .collection('Learing')
                .doc(widget.lessonId)
                .get();

            await favRef.set({
                'lessonId': widget.lessonId,
                'title': lessonDoc['title'] ?? '',
                'timestamp': FieldValue.serverTimestamp(),
            });
        }
    }

    if (mounted) {
        setState(() {
            isFavorite = !isFavorite;
        });
    }
} catch (e) {
    debugPrint('Error toggling favorite: $e');
}
}

@Override
Widget build(BuildContext context) {
    return IconButton(
        icon: Icon(
            isFavorite ? Icons.favorite : Icons.favorite_border,
            color: Colors.green[800],
        ), // Icon
        onPressed: toggleFavorite,
    ); // IconButton
}
```

Figure 56: `toggleFavorite()`

Semester-2 1446H (Spring 2025)

▪ Toggling Video Favorites

The favorite video icon feature in the app is encapsulated by the `FavoriteVideoIcon` widget, which takes `lessonId`, `subtitle`, and `URL` as parameters. In its state class,

`_FavoriteVideoIconState`, a Boolean `isFavorite` flag is initialized to false and the current user's ID is obtained from `FirebaseAuth`. In `initState()`, it immediately calls the asynchronous `checkIfFavorite()` method. This method constructs a `Firestore` reference to `Favorites/{userId}/lessons/{lessonId}/videos/{subtitle}` and attempts to fetch that document. If the document exists, it sets `isFavorite = true` inside `setState()`, causing the heart icon to render filled. Any errors during this read are caught and logged with `debugPrint()`, ensuring the UI remains responsive even if the lookup fails.

```
class FavoriteVideoIcon extends StatefulWidget {
    final String lessonId;
    final String subtitle;
    final String url;

    const FavoriteVideoIcon({
        required this.lessonId,
        required this.subtitle,
        required this.url,
    });

    @override
    _FavoriteVideoIconState createState() => _FavoriteVideoIconState();
}

class _FavoriteVideoIconState extends State<FavoriteVideoIcon> {
    bool isFavorite = false;
    final userId = FirebaseAuth.instance.currentUser?.uid;

    @override
    void initState() {
        super.initState();
        checkIfFavorite();
    }

    Future<void> checkIfFavorite() async {
        if (userId == null) return;
        final doc = await FirebaseFirestore.instance
            .collection('Favorites')
            .doc(userId)
            .collection('lessons')
            .doc(widget.lessonId)
            .collection('videos')
            .doc(widget.subtitle)
            .get();

        setState(() {
            isFavorite = doc.exists;
        });
    }
}
```

Figure 57: `FavoriteVideoIconState`

Semester-2 1446H (Spring 2025)

Tapping the heart icon invokes the asynchronous `toggleFavorite()` method, which first guards against a null `userId`. It then builds two Firestore references: one to the lesson document at `Favorites/{userId}/lessons/{lessonId}` and one to its video subcollection at `.../videos/{subtitle}`. If `isFavorite` is true, it deletes the video document from Firestore; otherwise, it first checks whether the parent lesson document exists, if not, it fetches its title from `Learing/{lessonId}` and creates that document with a title field—then writes a new video document containing subtitle, URL, and `lessonId`. After the write or delete completes, it flips `isFavorite` inside `setState()` so the icon updates immediately. All exceptions during these operations are caught and printed with `debugPrint()`, preserving a smooth user experience.

```

    Future<void> toggleFavorite() async {
        if (userId == null) return;

        final lessonDocRef = FirebaseFirestore.instance
            .collection('Favorites')
            .doc(userId)
            .collection('Lessons')
            .doc(widget.lessonId);

        final videoDocRef =
            lessonDocRef.collection('videos').doc(widget.subtitle);

        if (isFavorite) {
            await videoDocRef.delete();
        } else {
            final lessonExists = await lessonDocRef.get();
            if (!lessonExists.exists) {
                final lessonData = await FirebaseFirestore.instance
                    .collection('Learing')
                    .doc(widget.lessonId)
                    .get();
                final title = lessonData.data()?['title'] ?? '';
                await lessonDocRef.set({'title': title});
            }
        }

        await videoDocRef.set({
            'subtitle': widget.subtitle,
            'url': widget.url,
            'lessonId': widget.lessonId,
        });
    }

    setState(() {
        isFavorite = !isFavorite;
    });
}

@Override
Widget build(BuildContext context) {
    return IconButton(
        icon: Icon(
            isFavorite ? Icons.favorite : Icons.favorite_border,
            color: Colors.green,
        ), // Icon
        onPressed: toggleFavorite,
    ); // IconButton
}

```

Figure 58: `toggleFavorite()`

12. External APIs Utilized

We enriched Nomu with an AI-powered chatbot by integrating Google's Gemini Generative Language API, giving users instant, context-aware answers to any trading or finance question.

- **Dependencies**

Before any code, we added two key packages in pubspec.yaml :

```
dependencies:  
  http: ^1.2.1 # For making REST calls to Gemini  
  flutter_dotenv: ^5.1.0 # For securely loading GEMINI_API_KEY from a .env file
```

Figure 59: Dependencies & Configuration

we securely load the Gemini API key from a .env file at the project root containing the API key, by initializing flutter_dotenv in main().

```
Future<void> main() async {  
  // Ensure Flutter bindings are initialized (Good!)  
  WidgetsFlutterBinding.ensureInitialized();  
  
  // Load Environment Variables for the API Key  
  try {  
  
    await dotenv.load(fileName: ".env");  
    print("DotEnv loaded successfully.");  
  } catch (e) {  
    print("Error loading .env file: $e");  
  }  
}
```

Figure 60: load the API key

Semester-2 1446H (Spring 2025)

We then encapsulate all of the AI logic in a single GeminiService method, Future<String?> generateText(String prompt, List<ChatMessage> history), which reads the key via dotenv.env['GEMINI_API_KEY'], wraps the prompt and chat history into Gemini's required JSON format, sends it in an HTTP POST (using the http package), and parses out the model's text reply. By centralizing environment-variable management, network communication, and JSON parsing in this one method, our UI layer only needs to call generateText and render its result.

```

    < class GeminiService {
        final String? _apiKey = dotenv.env['GEMINI_API_KEY'];
        final String _model = "gemini-1.5-flash";
        final String _apiVersion = "v1beta";

        < Future<String?> generateText(String prompt, List<ChatMessage> history) async {
            if (_apiKey == null) return "Error: API Key not configured.";

            < final url = Uri.parse(
                "https://generativelanguage.googleapis.com/${_apiVersion}/models/${_model}:generateContent?key=$_apiKey"
            );

            < final apiHistory = history.map((msg) => {
                "role": msg.isUserMessage ? "user" : "model",
                "parts": [{"text": msg.text}]
            }).toList();

            < final contentPayload = [
                ...apiHistory,
                {"role": "user", "parts": [{"text": prompt}]}
            ];

            < final body = json.encode({"contents": contentPayload});

            < final response = await http.post(
                url,
                headers: {'Content-Type': 'application/json'},
                body: body,
            );

            < if (response.statusCode == 200) {
                final data = json.decode(response.body);
                final candidates = data['candidates'];
                if (candidates != null && candidates.isNotEmpty) {
                    return candidates[0]['content']['parts'][0]['text'];
                }
                return "";
            } else {
                return "Error: ${response.statusCode}";
            }
        }
    }
}

```

Figure 61: GeminiService

7.6 Challenges

During the development of the project, we encountered several key challenges that required thoughtful solutions and iterative testing:

1. Displaying YouTube videos inside the app:

One of the biggest challenges was embedding YouTube videos within the app instead of opening them in an external application. We solved this by using the youtube_player_flutter package and customizing the video player to match our app's design and provide a smooth in-app viewing experience.

2. Storing and managing coins accurately:

Implementing the reward system for coins required careful handling to ensure that coins are added only once when the user answers correctly and are stored reliably in Firebase Firestore.

3. “Forgot Password” feature:

Initially, the password reset functionality followed Google's password rules, but some passwords considered strong were being flagged as weak by Firebase. To resolve this, we adjusted the password validation logic to better align with our app's structure while maintaining security standards.

4. Flutter project setup and dependency delays:

Another technical challenge was the time-consuming setup process for the Flutter environment. On devices with average performance or slow internet connections, downloading dependencies and building the app for the first time took considerable time.

5. Building AI-generated Saudi-accented voice for videos:

Creating natural-sounding educational voiceovers in Arabic presented a significant technical challenge, as standard text-to-speech engines often lack the nuance required for regional dialects. To address this, we conducted a comparative analysis of multiple AI voice

Semester-2 1446H (Spring 2025)

providers, including ElevenLabs[32], NabaRati[33], and Play.ht[9]. While ElevenLabs offered expressiveness, its output was limited to formal Modern Standard Arabic, and NabaRati often sounded mechanical; consequently, we selected Play.ht (PlayAI) for its superior balance of realistic voice variety and deep customization options. Beyond platform selection, our team engaged in rigorous linguistic engineering to localize the output to an engaging Saudi accent. This process involved manually inserting diacritics (Tashkeel) into the script, such as writing "سوق" instead of "سُوق" to force the correct pronunciation of "market", and utilizing the platform's phonetic editor to adjust vowel sounds to match local speech patterns. We further refined the audio by slowing the narrator's pace for clarity and adjusting the pitch to create a welcoming, conversational tone, ensuring the final content was both culturally relevant and highly effective for learners.

6. Prediction model loading delays:

Integrating the AI prediction module introduced noticeable loading delays, especially when fetching both historical prices and prediction documents simultaneously. Because each company required multiple Firestore reads, users occasionally experienced slow initialization and longer waiting times. To address this, we implemented staged loading through the `_bootstrapPage()` method, added rotating status messages to improve perceived responsiveness, and introduced caching to minimize repeated data retrieval.

7. Slow loading of market charts and top-performer data:

Rendering the Market Overview Chart, Company Chart, and the Top Five weekly performers introduced noticeable delays due to the large volume of historical price data required for each component. Fetching multiple Firestore documents, computing weekly changes, and rebuilding chart series—especially when several companies were selected—led to slower response times on mid-range devices. To address this, we implemented data caching, optimized Firestore queries, and added Firestore composite indexes to speed up repeated lookups. These improvements significantly reduced loading times and provided a smoother user experience.

8. Unifying data consistency across all market modules:

One of the major challenges during development was ensuring that all market-related components—such as the Market Overview Chart, Company Chart, Top Five performers, Prediction Model, and Portfolio calculations—used consistent and synchronized data. Because each module fetched historical prices independently and applied different date mappings, slight mismatches sometimes appeared in values and timestamps. To resolve this, we standardized the date-conversion process, unified the Firestore access layer, and introduced shared caching so all components would reference the same data sources. This alignment improved accuracy, eliminated discrepancies, and ensured a coherent user experience across the entire simulation system.

9. Ensuring reliable delivery and control of external notifications:

A key challenge we encountered was managing external push notifications delivered through Firebase Cloud Messaging (FCM) and ensuring that they consistently reached the user. Since FCM delivery can be influenced by device network conditions and Android background restrictions, guaranteeing timely and reliable delivery required additional validation. Another significant challenge was providing users with clear control over enabling or disabling FCM alerts. To address this, we integrated the application directly with the Android system notification settings, allowing users to grant or revoke notification permissions at the OS level. This required synchronizing our in-app preferences with system-level permissions and verifying that the FCM token remained active. These measures helped ensure that external notifications were both permitted and successfully received by the user.

7.7 GitHub Repository

The complete source code for the Nomu Application is available at the following GitHub link:
<https://github.com/asmaib/GP-Nomu.git>

We crafted high-fidelity mockups that adhere to right-to-left reading conventions and established a unified visual language. A persistent five-tab navigation bar was defined to grant instant access to the home, learning, simulation, portfolio, and More screens. We standardized colors, typography, and card layouts across all key components including educational content, daily

Semester-2 1446H (Spring 2025)

challenges, trading interfaces, market charts, prediction panels, and portfolio overviews to achieve a consistent and polished user experience on different screen sizes.

On the server side, Firebase Authentication was configured to manage all aspects of user identity—including signup, login, logout, password resets, and profile updates—while Firestore served as the main data store for the application. We structured collections and subcollections for users, lessons, flashcards, earned-coin records, favorites, historical market prices, prediction results, trading orders, portfolio summaries, wallet balances, and notifications. Security rules were implemented to restrict access based on authentication state and indexed queries were added to optimize reads for charts, portfolio reconstruction, and prediction lookups. Offline persistence was enabled to maintain usability and data consistency even under unstable network conditions.

1. Client-Side Assembly

Using Flutter, we built modular screens and widgets to implement the UI prototypes. The onboarding and authentication flows include form validation and smooth RTL transitions. Video tutorials are streamed through the `youtube_player_flutter` plugin, with lesson URLs retrieved from Firestore. Flashcards use the `swipe_cards` and `flip_card` packages to enable a two-pass review experience with color-coded feedback. Daily challenges are selected randomly, stored per user in `shared_preferences`, and displayed through a custom dialog with Arabic radio buttons. A centralized `CoinManager` grants points exactly once per learning activity and records them in Firestore, while a dedicated Favorites module allows users to bookmark lessons, videos, or cards.

To support the investment features, we integrated historical market data across multiple modules. The simulation system includes a Market Overview Chart with dynamic filtering, CompanyChart screens with buy/sell actions, and a TradingService that records transactions and updates wallet balances. The prediction module displays actual versus predicted prices with clear recommendations. Portfolio logic reconstructs daily performance by replaying the user's trades, while the Wallet module manages balance resets and synchronization with Firestore. Local reminders and external FCM notifications are handled through a unified notification service that manages permissions, scheduling, and message delivery. Together, these components form a cohesive, responsive, and interactive experience aligned with the project's educational goals.

2. Quality Assurance and System Testing

We applied multiple layers of testing to ensure the reliability of all learning and simulation features. Unit tests validated key logic such as video loading, flashcard progression, daily challenge selection, coin awards, trading execution, wallet updates, portfolio calculations, prediction comparisons, and Firestore data handling. Widget tests were used to verify essential user interactions across lessons, playlists, flashcards, challenges, market charts, and trading screens.

Semester-2 1446H (Spring 2025)

After component-level validation, we performed comprehensive system testing to confirm correct end-to-end behavior across authentication, video learning, flashcards, daily challenges, simulation charts, prediction model, trading engine, portfolio tracking, wallet management, favorites, and notification services. These tests ensured accurate data flow, stable chart performance, consistent historical-data synchronization, and reliable delivery and logging of external FCM notifications. This combined testing process confirmed that the application operates smoothly and consistently across all features.

```
firebase_core: ^3.12.0
firebase_auth: ^5.5.0
cloud_firestore: ^5.6.4
1. firebase_dynamic_links: ^6.1.3
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(MyApp());
}
```

Setting up Firestore

Data:

```
UserCredential userCredential = await _auth.createUserWithEmailAndPassword(
  email: email,
  password: password,
);
await FirebaseFirestore.instance
  .collection('users')
  .doc(userCredential.user!.uid)
  .set({
  'username': username,
  'email': email,
  'coins': 0,
  'challenge_answered': false,
  'challenge_timestamp': Timestamp.fromDate(defaultTimestamp),
});

try {
  await _auth.signInWithEmailAndPassword(email: email, password: password);
} try {
  await _auth.signInWithEmailAndPassword(email: email, password: password);

  Navigator.pushReplacement(
    context,
    MaterialPageRoute(builder: (context) => HomePage()),
  );
}
```

Semester-2 1446H (Spring 2025)

```

} on FirebaseAuthException catch (_) {
    _showMessageDialog(
        'خطأ',
        '، عفواً، البيانات المدخلة غير صحيحة. يرجى التأكد من صحة البريد الإلكتروني وكلمة المرور والمحاولة مرة أخرى');
} catch (_) {
    _showMessageDialog(
        'خطأ',
        '، عفواً، حدث خطأ أثناء عملية تسجيل الدخول. يرجى المحاولة مرة أخرى');
}

try {
    await _auth.sendPasswordResetEmail(email: email);
    _showMessageDialog(
        'نجاح',
        '، إذا كان هناك حساب مسجل بهذا البريد الإلكتروني، فقد تم إرسال رابط لإعادة تعيين كلمة المرور',
        success: true, // Display the login button
    );
    setState(() {
        _linkSent = true;
    });
} on FirebaseAuthException catch (e) {
    if (e.code == 'invalid-email') {
        setState(() {
            _errorMessage = 'يرجى إدخال بريد إلكتروني صحيح';
        });
    }
}

if (shouldLogout == true) {
    await FirebaseAuth.instance.signOut();
    Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => LoginPage()));
}
}

try {
    await user.updateDisplayName(newName);

    await FirebaseFirestore.instance.collection('users').doc(user.uid).update({
        'username': newName,
    });
}

youtube_player_flutter: ^9.1.1
void initState() {
    super.initState();
    final videoId = YoutubePlayer.convertUrlToId(widget.videoUrl)!;
    _controller = YoutubePlayerController(
        initialVideoId: videoId,
        flags: YoutubePlayerFlags(
            autoPlay: true,
            mute: false,
        ), // YoutubePlayerFlags
    ); // YoutubePlayerController
}

swipe_cards: ^2.0.0+1
flip_card: ^0.6.0

```

Figure 30 Flash card 1

Semester-2 1446H (Spring 2025)

```
Future<void> fetchFlashcardsFromFirestore() async {
    try {
        final snapshot = await FirebaseFirestore.instance
            .collection('Learing')
            .doc(widget.lessonId)
            .collection('flashcard')
            .orderBy(FieldPath.documentId)
            .get();
    }

    Future<void> _initializeDailyQuestion() async {
        _todaysQuestion = await _getDailyQuestion();
    }

    Future<void> _markChallengeAnswered() async {
        final user = FirebaseAuth.instance.currentUser;
        if (user != null) {
            final now = DateTime.now();
            await FirebaseFirestore.instance.collection('users').doc(user.uid).update({
                'challenge_answered': true,
                'challenge_timestamp': Timestamp.fromDate(now),
            });
            setState(() {
                _challengeAnswered = true;
            });
        }
    }
}
```

Semester-2 1446H (Spring 2025)

كلية علوم الحاسوب والمعلومات
 كلية تقنية المعلومات

```

Future<void> _loadChallengeStatus() async {
final user = FirebaseAuth.instance.currentUser;
if (user != null) {
    final userDoc = await FirebaseFirestore.instance.collection('users').doc(user.uid).get();
    if (userDoc.exists && userDoc['challenge_answered'] != null) {
        final challengeAnswered = userDoc['challenge_answered'] ?? false;
        final timestamp = (userDoc['challenge_timestamp'] as Timestamp?)?.toDate();

        if (challengeAnswered && timestamp != null) {
            final difference = DateTime.now().difference(timestamp).inHours;
            if (difference >= 24) {
                // Reset challenge if more than 24 hours passed
                await FirebaseFirestore.instance.collection('users').doc(user.uid).update({
                    'challenge_answered': false,
                });
                setState(() {
                    _challengeAnswered = false;
                });
            } else {
                setState(() {
                    _challengeAnswered = true;
                });
            }
        } else {
            setState(() {
                _challengeAnswered = false;
            });
        }
    }
}

import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

class CoinManager {
    // Get the current coin balance for the logged-in user
    static Future<int> getCoins() async {
        final user = FirebaseAuth.instance.currentUser;
        if (user != null) {
            final doc = await FirebaseFirestore.instance.collection('users').doc(user.uid).get();
            return doc.data()['coins'] ?? 0; // Default to 0 if no coins field exists
        }
        return 0; // Return 0 if no user is logged in
    }

    // Add coins to the user's balance
    static Future<void> addCoin(int amount) async {
        final user = FirebaseAuth.instance.currentUser;
        if (user != null) {
            final userRef = FirebaseFirestore.instance.collection('users').doc(user.uid);
            await userRef.update({'coins': FieldValue.increment(amount)});
        }
    }

    // Initialize the user's coin balance if it's their first time logging in
    static Future<void> initializeCoins() async {
        final user = FirebaseAuth.instance.currentUser;
        if (user != null) {
            final userDoc = await FirebaseFirestore.instance.collection('users').doc(user.uid);
            if (!userDoc.exists) {
                final userRef = FirebaseFirestore.instance.collection('users').doc(user.uid);
                final userDoc = await userRef.get();
                if (!userDoc.exists) {
                    await userRef.set({
                        'coins': 0, // Start with 0 coins
                    });
                }
            }
        }
    }
}

```

Figure 31: coin manager

Semester-2 1446H (Spring 2025)

```
38     // Check if the user has earned coins for a specific flashcard lesson
39     static Future<bool> hasEarnedCoinsForLesson(String lessonId) async {
40         final user = FirebaseAuth.instance.currentUser;
41         if (user != null) {
42             final doc = await FirebaseFirestore.instance
43                 .collection('users')
44                 .doc(user.uid)
45                 .collection('earnedFlashcardCoins')
46                 .doc(lessonId)
47                 .get();
48             return doc.exists;
49         }
50         return false;
51     }
52
53     // Mark that the user has earned coins for a specific flashcard lesson
54     static Future<void> markCoinsAsEarnedForLesson(String lessonId) async {
55         final user = FirebaseAuth.instance.currentUser;
56         if (user != null) {
57             final docRef = FirebaseFirestore.instance
58                 .collection('users')
59                 .doc(user.uid)
60                 .collection('earnedFlashcardCoins')
61                 .doc(lessonId);
62             await docRef.set({'earned': true});
63         }
64     }
65 }
```

Semester-2 1446H (Spring 2025)

```
82      Future<void> checkAndRewardCoins() async {
83        try {
84          final userId = FirebaseAuth.instance.currentUser?.uid;
85          if (userId == null || !mounted) return;
86
87          final userCoinsRef = FirebaseFirestore.instance
88              .collection('users')
89              .doc(userId)
90              .collection('lessons_rewards')
91              .doc(widget.lessonDocId);
92
93          final userCoinsDoc = await userCoinsRef.get();
94
95          if (!userCoinsDoc.exists) {
96            await userCoinsRef.set({
97              'coins': 10,
98              'timestamp': FieldValue.serverTimestamp()
99            });
100
101         final userRef = FirebaseFirestore.instance.collection('users').doc(userId);
102         await userRef.update({
103           'coins': FieldValue.increment(10),
104         });
105
106         if (mounted) {
107           showDialog(
108             context: context,
109             builder: (_) => AlertDialog(
110               title: const Text('مبروك'),
111               content: const Text('لتحت 10 عملات إضافية الى حسابك'),
112               actions: [
113                 TextButton(
114                   onPressed: () => Navigator.of(context).pop(),
115                   child: const Text('حسناً'),
116                 ),
117               ],
118             ),
119           );
120         }
121       }
122     } catch (e) {
123       debugPrint('Error rewarding coins: $e');
124     }
125   }
126 }
```

1.

Semester-2 1446H (Spring 2025)

```
44     Future<void> checkAndRewardCoins() async {
45         final userId = FirebaseAuth.instance.currentUser?.uid;
46         if (userId == null) return;
47
48         final userCoinsRef = FirebaseFirestore.instance
49             .collection('users')
50             .doc(userId)
51             .collection('lessons_rewards')
52             .doc(widget.videoTitle);
53
54         final userCoinsDoc = await userCoinsRef.get();
55
56         if (!userCoinsDoc.exists) {
57             await userCoinsRef.set({'coins': 10});
58
59             final userRef = FirebaseFirestore.instance.collection('users').doc(userId);
60             await userRef.update({
61                 'coins': FieldValue.increment(10),
62             });
63
64         if (mounted) {
65             showDialog(
66                 context: context,
67                 builder: (_) => AlertDialog(
68                     title: Text('مبروك'),
69                     content: Text('لتحت 10 عملات لفافية الى صندوقك'),
70                     actions: [
71                         TextButton(
72                             onPressed: () {
73                                 Navigator.of(context).pop();
74                             },
75                             child: Text('حسناً'),
76                         ),
77                     ],
78                 ),
79             );
80         }
81     }
82 }
```

Semester-2 1446H (Spring 2025)

```
FutureBuilder<bool>(
    future: CoinManager.hasEarnedCoinsForLesson(lessonId),
    builder: (context, snapshot) {
        if (snapshot.connectionState != ConnectionState.done) {
            return CircularProgressIndicator(color: Colors.green);
        }

        final hasEarned = snapshot.data ?? false;
        final message = hasEarned
            ? "ممتاز! أنت الآن جاهز لبدء رحلة الاستئتمار"
            : "تهانينا! تمت إضافة 10 عملات إلى مصبيك \n كل خطوة تعلم تقنيك نحو النجاح!";

        return Text(
            message,
            textAlign: TextAlign.center,
            style: TextStyle(
                fontSize: 22,
                color: Colors.black87,
                fontWeight: FontWeight.bold,
            ), // TextStyle
        ); // Text
    },
), // FutureBuilder
SizedBox(height: 40),
ElevatedButton(
    onPressed: () async {
        bool hasEarned = await CoinManager.hasEarnedCoinsForLesson(lessonId);
        if (!hasEarned) {
            await CoinManager.addCoin(10);
            await CoinManager.markCoinsAsEarnedForLesson(lessonId);
        }
        Navigator.pop(context, true);
    },
    child: Text("العودة للدروس", style: TextStyle(fontSize: 16)),
)
```

Semester-2 1446H (Spring 2025)

```
void _showCorrectDialog() {
    showDialog(
        context: context,
        barrierDismissible: false,
        builder: (ctx) => Stack(children: [
            SizedBox.expand(child: Lottie.asset('assets/fireworks.json', fit: BoxFit.cover, repeat: true)),
            Center(
                child: Column(mainAxisSize: MainAxisSize.min, children: [
                    Text("مُنْجَاحٌ", style: TextStyle(fontSize: 40, color: Colors.white, fontWeight: FontWeight.bold)),
                    SizedBox(height: 30),
                    ElevatedButton(
                        onPressed: () async {
                            final user = FirebaseAuth.instance.currentUser;
                            if (user != null) {
                                await FirebaseFirestore.instance.collection('users').doc(user.uid)
                                    .update({'coins': FieldValue.increment(5)});
                            }
                            Navigator.pushReplacement(context, MaterialPageRoute(builder: (_) => HomePage()));
                            _markChallengeAnswered();
                        },
                        style: ElevatedButton.styleFrom(backgroundColor: Color(0xFF9DC08B)),
                        child: Text("إغلاق"),
                    ) // ElevatedButton
    ])));
}
```

Market Data Integration and Simulation Engine

- Market Overview Chart Logic

The Market Overview Chart provides interactive multi-company visualization that helps users gain insight into broad market movements. The bottom-sheet filtering interface allows users to select up to five companies using dynamically generated checkbox tiles. When the user confirms their selection, the page resolves the corresponding company IDs, preloads missing historical data through `_loadCompanyDataToCache()`, and rebuilds the entire chart by calling `_buildSeriesFromCache()`. This ensures efficient updates and accurate plotting at all times.

To support flexible market analysis, the chart integrates custom `TfChip` widgets that switch between different time windows such as 1W, 1M, and 3M. These chips animate selections and apply Nomu's green color theme for consistency across the simulation module.

Together, these components provide a highly responsive and intuitive charting experience tailored to Arabic RTL users.

Figure 54 shows the filtering logic, **Figure 55** displays the chip implementation, and **Figure 56** illustrates the chart-rebuild workflow.

Semester-2 1446H (Spring 2025)

```
ElevatedButton(  
    onPressed: () async {  
        Navigator.pop(context);  
  
        // Find IDs for selected names and Load Data if missing  
        final selectedIds = <String>[];  
        _companyMeta.forEach((id, meta) {  
            if (tmpSelected.contains(meta['name'])) selectedIds.add(id);  
        });  
  
        // Show loading indicator in parent if needed, or just await  
        await Future.wait(selectedIds.map((id) => _loadCompanyDataToCache(id)));  
  
        if (mounted) {  
            setState(() {  
                _selectedCompanies..clear()..addAll(tmpSelected);  
                _future = Future.value(_buildSeriesFromCache(_window)); // Rebuild  
                _zoomPanBehavior.reset();  
            });  
        }  
    },  
    child: const Text('البحث'),  
,
```

Figure 32:filtering logic

Semester-2 1446H (Spring 2025)

```
class _TfChip extends StatelessWidget {
    final String label;
    final bool selected;
    final VoidCallback onTap;
    const _TfChip({required this.label, required this.selected, required this.onTap});

    @override
    Widget build(BuildContext context) {
        const color = Color(0xFF609966);

        return GestureDetector(
            onTap: onTap,
            child: AnimatedContainer(
                duration: const Duration(milliseconds: 150),
                padding: const EdgeInsets.symmetric(horizontal: 12, vertical: 6),
                decoration: BoxDecoration(
                    color: selected ? color.withOpacity(0.12) : Colors.grey.shade200,
                    borderRadius: BorderRadius.circular(999),
                    border: Border.all(
                        color: selected ? color : Colors.grey.shade300,
                        width: selected ? 1.6 : 1),
                ),
                child: Text(label,
                    style: TextStyle(
                        fontWeight: FontWeight.w600,
                        color: selected ? color : Colors.black87)),
            ),
        );
    }
}
```

Figure 33:the chip implementation

```
Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
        _TfChip(label: '1W', selected: _window == MarketWindow.w1, onTap: () => _setWindow(MarketWindow.w1)),
        const SizedBox(width: 8),
        _TfChip(label: '1M', selected: _window == MarketWindow.m1, onTap: () => _setWindow(MarketWindow.m1)),
        const SizedBox(width: 8),
        _TfChip(label: '3M', selected: _window == MarketWindow.m3, onTap: () => _setWindow(MarketWindow.m3)),
    ],
),
```

Figure 34:the chart-rebuild workflow

■ Top Five Weekly Performers Logic

The simulation module highlights the strongest market movers through a “Top Five Weekly Performers” section. This feature computes weekly gains by comparing each company’s most recent price record with its value at the beginning of the simulated week. Using historical data stored in

Semester-2 1446H (Spring 2025)

Firestore, the algorithm calculates percentage change for every company, sorts them in descending order, and selects the top five performers.

Each company is presented in a stylized RTL card containing its logo, name, current price, and weekly percentage gain. Tapping any entry navigates directly to its detailed chart in the CompanyChartWidget, creating a seamless transition between market ranking and in-depth analysis. This feature helps learners quickly identify high-performing stocks and explore potential patterns inside the simulation environment.

Figures 57-59 shows the weekly-gainer calculation.

```
Future<void> _fetchTopGainersWTD_StringQuery() async {
  if (simulationStartRealDate == null) return;
  if (_cachedWeeklyGainers != null && _cachedWeeklyGainers!.isNotEmpty) return;

  setState(() {
    _isLoadingGainers = true;
    _loadingPercent = 0.0;
    _loadingProgressText = "0%";
  });
  _startLoadingMessages();

  try {
    final currentSimDate = _calculateSimulatedDateWithCloseRule();

    // Convert simulation date to String "YYYY-MM-DD" for Firebase comparison
    final String simDateStr = _formatDateForDB(currentSimDate);

    final firestore = FirebaseFirestore.instance;
    final companiesSnap = await firestore.collection('companies').get();
    final totalCompanies = companiesSnap.docs.length;

    List<Gainer> allResults = [];
    int processedCount = 0;
    int batchSize = 20;

    for (var i = 0; i < totalCompanies; i += batchSize) {
      final end = (i + batchSize < totalCompanies) ? i + batchSize : totalCompanies;
      final batch = companiesSnap.docs.sublist(i, end);

      await Future.wait(batch.map((cDoc) async {
        final data = cDoc.data();
        final name = (data['name'] ?? cDoc.id).toString();
        final logo = (data['LogoAsset'] ?? '').toString();
        final gainer = Gainer(name: name, logo: logo);
        allResults.add(gainer);
        processedCount++;
      }));
    }
  }
}
```

Figure 35: weekly-gainer calculation 1

Semester-2 1446H (Spring 2025)

```

try {
    // 🔞 QUERY FIX: Using String Comparison
    // "Give me the last 5 days ON OR BEFORE the simulation date"
    final snapshot = await cDoc.reference
        .collection('PriceRecords_full')
        .where('date', isLessThanOrEqualTo: simDateStr) // String compare works for ISO dates
        .orderBy('date', descending: true)
        .limit(5)
        .get();

    if (snapshot.docs.isNotEmpty) {
        // Latest price (The record closest to simulation date)
        final endPrice = (snapshot.docs.first.data()['close'] as num).toDouble();

        // Oldest price in the batch (up to 5 days ago)
        // If we have 5 days, use the 5th. If we only have 2, use the 2nd.
        final startPrice = (snapshot.docs.last.data()['close'] as num).toDouble();

        if (startPrice > 0 && endPrice > 0) {
            // If only 1 record exists (new company), change is 0.0
            final changePct = (snapshot.docs.length < 2)
                ? 0.0
                : ((endPrice - startPrice) / startPrice) * 100.0;

            // Keep everything so the list isn't empty, logic handles sorting
            allResults.add(_Gainer(
                companyId: cDoc.id,
                name: name,
                logoAsset: logo,
                lastPrice: endPrice,
                changePct: changePct,
            ));
        }
    }
}

```

Figure 36:weekly-gainer calculation2

```

    } catch (...) {};
});

processedCount += batch.length;
if (mounted) {
    setState(() {
        _loadingPercent = processedCount / totalCompanies;
        _loadingProgressText = "${(processedCount / totalCompanies * 100).toInt()}%";
    });
}
await Future.delayed(const Duration(milliseconds: 1));
}

// Sort: Highest gains first
allResults.sort((a, b) => b.changePct.compareTo(a.changePct));

if (mounted) {
    setState(() {
        // Take Top 5. If list is empty, UI will show "No Data" gracefully.
        _cachedWeeklyGainers = allResults.take(5).toList();
        _isLoadingGainers = false;
    });
}
} catch (e) {
    debugPrint("Error loading gainers: $e");
    if (mounted) setState(() => _isLoadingGainers = false);
} finally {
    _msgTimer?.cancel();
}
}

```

Figure 37:weekly-gainer calculation3

2. CompanyChart & Trading Logic

When a user selects a company from the simulation page, the system loads its historical price records and plots them into an RTL chart optimized for Arabic learners. The `_convertToDate()` method aligns simulated trading days with real Tadawul market days by skipping Fridays and Saturdays, ensuring that all plotted data reflects authentic Saudi trading behavior.

Users can initiate a transaction directly from this screen. When the Buy or Sell button is pressed, the widget displays a localized Arabic confirmation sheet that summarizes the order using `_buildReceiptRow()`, which formats details such as the stock symbol, selected quantity, unit price, and total value. Confirming the trade triggers the `TradingService.placeTrade()` function, which records the transaction under the user's Firestore orders collection, updates open positions, and recalculates wallet balances accordingly. This workflow guarantees that every buy or sell action immediately impacts the user's simulated portfolio and wallet.

Figure 60 illustrates the date-conversion logic, **Figure 61** shows the Arabic receipt layout, and **Figure 62** displays the buy/sell transaction implementation.

Semester-2 1446H (Spring 2025)

```
DateTime _convertToRealDate(
    DateTime firebaseDate, List<DateTime> allTradingDays, DateTime simDate) {
    int indexInSim = allTradingDays.indexWhere((d) =>
        d.year == firebaseDate.year &&
        d.month == firebaseDate.month &&
        d.day == firebaseDate.day);
    if (indexInSim == -1) return firebaseDate;

    int currentIndex = allTradingDays.indexWhere((d) =>
        d.year == simDate.year &&
        d.month == simDate.month &&
        d.day == simDate.day);
    if (currentIndex == -1) return firebaseDate;

    int daysDiff = currentIndex - indexInSim;

    DateTime realToday = DateTime.now();
    if (realToday.weekday == DateTime.friday) {
        realToday = realToday.subtract(const Duration(days: 1));
    } else if (realToday.weekday == DateTime.saturday) {
        realToday = realToday.subtract(const Duration(days: 2));
    }

    DateTime resultDate = realToday;
    int tradingDaysSubtracted = 0;
    while (tradingDaysSubtracted < daysDiff) {
        resultDate = resultDate.subtract(const Duration(days: 1));
        if (resultDate.weekday != DateTime.friday &&
            resultDate.weekday != DateTime.saturday) {
            tradingDaysSubtracted++;
        }
    }
    return resultDate;
}
```

Figure 38:the date-conversion logic

```
Widget _buildReceiptRow(String label, String value, {Color? valueColor, bool isBold = false}) {
    return Padding(
        padding: const EdgeInsets.only(bottom: 8),
        child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
                Text(label, style: TextStyle(color: Colors.grey.shade600, fontSize: 13)),
                Text(
                    value,
                    style: TextStyle(
                        color: valueColor ?? Colors.black87,
                        fontWeight: isBold ? FontWeight.bold : FontWeight.w500,
                        fontSize: 14
                    ),
                ),
            ],
        ),
    );
}
```

Figure 39:the Arabic receipt layout

Semester-2 1446H (Spring 2025)

```

final cost = price * qty;

if (side == 'BUY') {
    if (cash < cost) {
        throw Exception('INSUFFICIENT_CASH');
    }
    // avg cost
    final newQty = currentQty + qty;
    final newAvg = (currentQty == 0) ? price : ((avgCost * currentQty) + cost) / newQty;

    // ensure normalized ref
    posRef = await _upsertPositionRef(vid, symbol: symbol, companyName: name);

    tx.set(
        posRef,
        {
            'symbol': symbol,
            'name': name,
            'qty': newQty,
            'avgCost': double.parse(newAvg.toStringAsFixed(6)),
            'updatedAt': FieldValue.serverTimestamp(),
        },
        SetOptions(merge: true),
    );

    tx.update(walletRef, {'cash': cash - cost});
} else {
    // SELL
    if (currentQty < qty) {
        throw Exception('INSUFFICIENT SHARES');
    }

    // ensure normalized ref
    posRef = await _upsertPositionRef(vid, symbol: symbol, companyName: name);

    final remain = currentQty - qty;
    if (remain == 0) {
        tx.delete(posRef);
    } else {
        tx.set(
            posRef,
            {
                'qty': remain,
                // keep same avgCost
                'avgCost': avgCost,
                'symbol': symbol,
                'name': name,
                'updatedAt': FieldValue.serverTimestamp(),
            },
            SetOptions(merge: true),
        );
    }

    tx.update(walletRef, {'cash': cash + cost});
}

```

Figure 40:the buy/sell transaction implementation.

3. Market Prediction Module

The Prediction Model feature displays AI-generated forecasts for each company based on pre-computed values stored in Firestore. When the page loads, the `_bootstrapPage()` method retrieves the current simulation date, loads the user's favorites, and fetches both actual historical prices and predicted values. During this process, the interface shows rotating status messages from `_loadingMessages` to guide the user through each loading stage.

For every company, the model compares the latest closing price with the predicted value and calculates the percentage difference. Based on this comparison, the system generates a simple recommendation **Buy** if the predicted value is higher, or **Don't Buy** otherwise. All results are presented in an RTL-friendly list, allowing users to quickly understand model-based signals without interacting with complex machine-learning details.

Figure 63 shows the bootstrap logic, and **Figure 64** illustrates the prediction comparison and recommendation rule.

```
Future<void> _bootstrapPage() async {
    // Stage 1: Load User & Favorites
    setState(() => _loadingProgressText = "20%");
    await _loadFavorites();

    // Stage 2: Calculate Simulation Date
    setState(() => _loadingProgressText = "45%");
    await _loadSimulationAnchorAndCompute();

    // Stage 3: Load Companies
    setState(() => _loadingProgressText = "70%");
    await _loadInitialCompanies();

    // Finish
    setState(() => _loadingProgressText = "100%");
}
```

Figure 41:the bootstrap logic

Semester-2 1446H (Spring 2025)

```
// Re-fetch prediction if we switched to a fallback date
if (!predDoc.exists && _cachedLatestAvailableId != null) {
    predDoc = await _withTimeout(predCol.doc(_cachedLatestAvailableId).get());
}

if (!priceDoc.exists || !predDoc.exists) {
    return {'insufficient': true, 'message': 'بيانات غير متوفرة'};
}

final latestClose = (priceDoc.data()?[['close']] as num?)?.toDouble() ?? 0.0;
final predicted = (predDoc.data()?[['predicted']] as num?)?.toDouble();

if (predicted == null || latestClose == 0) {
    return {'insufficient': true, 'message': 'بيانات غير متوفرة'};
}

final diff = predicted - latestClose;
final pct = (diff / latestClose) * 100.0;
final decisionAr = (predicted < latestClose) ? 'نحو الصعود' : 'نحو الصعود';

return {
    'close': latestClose,
    'predicted': predicted,
    'decisionAr': decisionAr,
    'pct': pct,
    'simDate': _currentSimDate!.toIso8601String(),
};
```

Figure 42:the prediction comparison and recommendation rule.

4. Portfolio Page (Performance Tracking)

The PortfolioPage reconstructs the user's simulated investment journey.

The `_loadChart()` method generates historical portfolio values by replaying past orders day by day, storing them in `_chartDays` and `_chartTotals` arrays.

`_calcPnLFromOrders()` iterates through all transactions, revaluing each open lot at historical closing prices and aggregating total profit, loss, and cost.

The final data feed is rendered via the `fl_chart` library as an animated RTL line chart showing performance over the selected period.

Figures 65–67 depict the chart computation and P&L calculation logic.

Semester-2 1446H (Spring 2025)

```

if (_range == '1D') {
    // ===== نداء اليوم (baseline + realized from today's sells) =====

    // اختر أحدث السهم وما قبل السهم
    final todayOrders = <QueryDocumentSnapshot<Map<String, dynamic>>>[];
    final beforeTodayOrders = <QueryDocumentSnapshot<Map<String, dynamic>>>[];
    for (final d in orders) {...}
    todayOrders.sort((a, b) => _extractDate(a.data()).compareTo(_extractDate(b.data())));

    // قيمة البالанс بـ الساعة (بيان بـ الساعة)
    final startRes = await _calcPnLFromOrders(beforeTodayOrders, ignoreSameDay: true);
    final currentValueStart = startRes.currentValue;

    // كاش بداية اليوم = الكاش الحالى - مصروف كاش اليوم
    double netCashChangeToday = 0.0;
    for (final d in todayOrders) {...}
    final cashAtStart = cash - netCashChangeToday;

    // تحرير FIFO من ما قبل السهم
    final fifoBySymbol = <String, List<Map<String, double>>>{};
    for (final lot in _buildOpenLotsFromOrders(beforeTodayOrders)) {...}

    final baselineTotal = cashAtStart + currentValueStart;

    final totals = <double>[baselineTotal];
    final days = <DateTime>[DateTime(todayReal.year, todayReal.month, todayReal.day, 9, 30)];

    double realizedToday = 0.0;

    for (final d in todayOrders) {...}

    // ---- تأكيد أن آخر نتيجة == إجمالي المحفظة الحالى (مثل الكشف)
    final totalNow = await _resolveTotalPortfolioNow(orders, cash);
    if (totals.isEmpty) {...} else {...}

    _chartTotals = totals;
    _chartDays = days;
} else {
    // 1W/1M/3M
}

```

Figure 43:the chart computation and P&L calculation logic1

Semester-2 1446H (Spring 2025)

```

    } else {
        // 1W/1M/3M
        final count = _range == '1W' ? 7 : (_range == '1M' ? 30 : 90);

        // جملة أيام العمل
        final days = <DateTime>[];
        DateTime cur = todayReal;
        while (days.length < count) {
            if (SimulationUtils.isWorkday(cur)) days.add(cur);
            cur = cur.subtract(const Duration(days: 1));
        }
        days.sort();

        // مجموعات
        final totals = <double>[];
        for (final d in days) {
            final v = await _portfolioAtDay(d, orders, cash, firstOrderReal);
            totals.add(v);
        }
        _chartDays = days;
        _chartTotals = totals;

        // ----- تأكيد أن آخر نتائج المخطط (بتاريخ النهاية) == المجموعات -----
        if (_chartTotals.isNotEmpty) {
            _chartTotals[_chartTotals.length - 1] = await _resolveTotalPortfolioNow(orders, cash);
        }
    }

    _chartLoading = false;
    setState(() {});
}

```

Figure 44:the chart computation and P&L calculation logic2

```

    Future<{double totalProfit, double totalLoss, double totalCost, double currentValue}> _calcPnLFromOrders(
    List<QueryDocumentSnapshot<Map<String, dynamic>>> orderDocs,
    bool ignoreSameDay = true,
) async {
    final simDatasetDay = await _effectivePricingDayNow();
    final openLots = _buildOpenLotsFromOrders(orderDocs);

    double totalProfit = 0.0;
    double totalLoss = 0.0;
    double totalCost = 0.0;
    double currentValue = 0.0;

    final Map<String, double> closeCache = {};

    for (final lot in openLots) {
        if (ignoreSameDay) {
            final lotDatasetDay = await _effectivePricingDayForBuy(lot.buyDate);
            if (_sameYMD(lotDatasetDay, simDatasetDay)) {
                continue; // إغلاق الموقف
            }
        }

        final symbol = lot.symbol;
        final close = closeCache.containsKey(symbol)
            ? closeCache[symbol]!
            : (await _latestCloseUntil(symbol, simDatasetDay) ?? 0.0);
        closeCache[symbol] = close;

        final nowVal = close * lot.qty;
        final cost = lot.price * lot.qty;
        final diff = nowVal - cost;

        totalCost += cost;
        currentValue += nowVal;

        if (diff > 0) totalProfit += diff;
        else if (diff < 0) totalLoss += -diff;
    }
    return {totalProfit: totalProfit, totalLoss: totalLoss, totalCost: totalCost, currentValue: currentValue};
}

```

Figure 45:the chart computation and P&L calculation logic3

5. Wallet Management

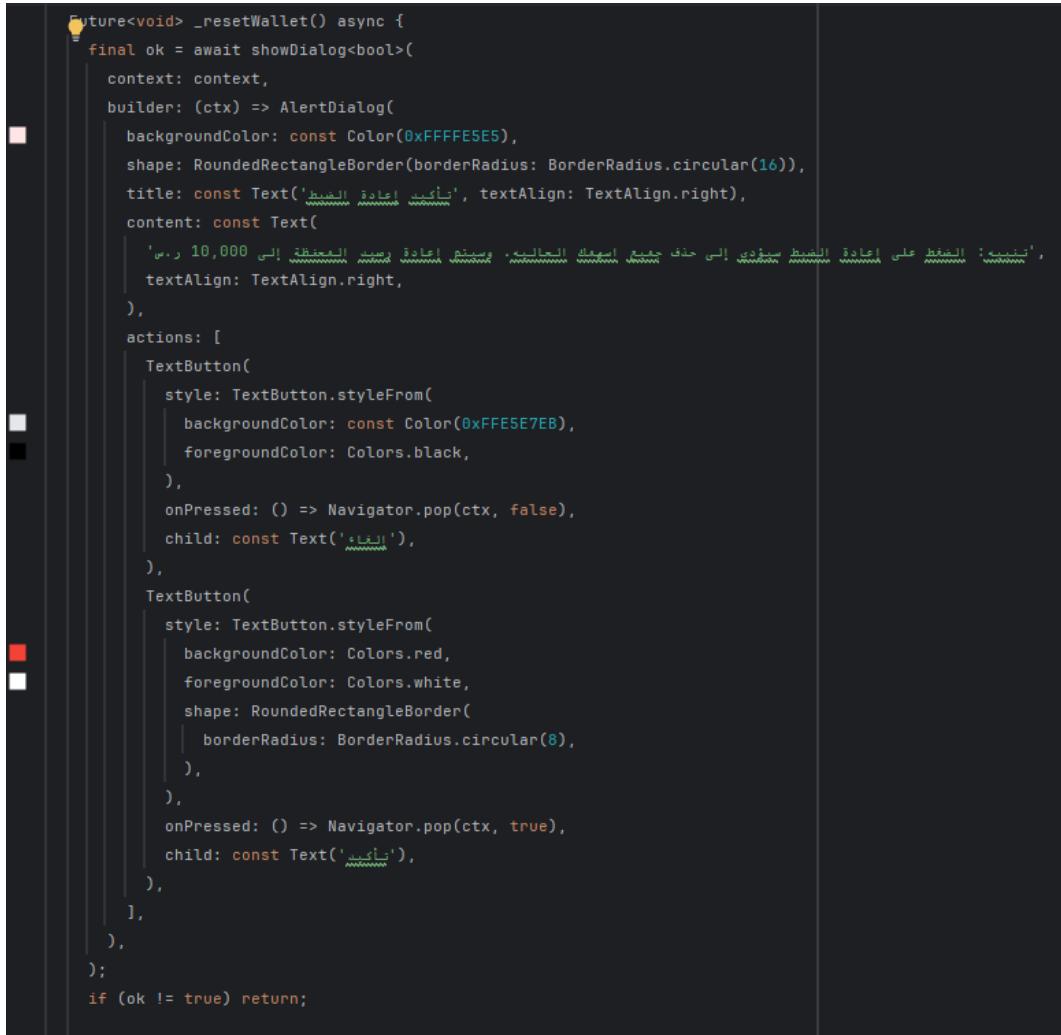
The WalletPage centralizes balance tracking and provides a controlled reset feature for testing. The _resetWallet() function displays a confirmation dialog (Figure 68) before restoring the virtual balance to 10 000 SAR.

Once confirmed, it rewrites the Firestore wallet documents (users/{uid}/wallet/main and Main) and purges the orders and positions subcollections to reinitialize the simulation.

Immediate feedback is shown through an RTL Snackbar notification.

This feature safeguards users from accidental data loss while supporting repeat training cycles.

Figures 68–69 highlight the confirmation prompt and reset execution logic.



A screenshot of a mobile application interface showing a confirmation dialog box. The dialog has a light blue background and contains the following text in Arabic:

تحذير: الخطأ على إعادة التعيين سيؤدي إلى حذف جميع أموالك الحالية. سيتم إعادة تعيين المحفظة إلى 10,000 روپ.

The dialog includes two buttons: "نعم" (Yes) in green and "لا" (No) in red. Below the buttons is a "Cancel" button.

Figure 46:the confirmation prompt

Semester-2 1446H (Spring 2025)

```
final uid = FirebaseAuth.instance.currentUser?.uid;
if (uid == null) return;

try {
    final walletCol = FirebaseFirestore.instance
        .collection('users')
        .doc(uid)
        .collection('wallet');

    // main/Main اعادة ضبط المكافأة إلى 10000 في كل من
    await walletCol.doc('main').set({'cash': _seed}, SetOptions(merge: true));
    await walletCol.doc('Main').set({'cash': _seed}, SetOptions(merge: true));

    // 10,000 حذف جميع الأدوات والتذاكر من شعب المخدم//الرسائل
    await _deleteUserSubcollection(uid, 'orders');
    await _deleteUserSubcollection(uid, 'positions');

    if (!mounted) return;
    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text("تمت إعادة ضبط المكافأة إلى 10,000 درهم")),
    );
} catch (e) {
    if (!mounted) return;
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('حدث خطأ في إعادة ضبط المكافأة $e')),
    );
}
```

Figure 47:reset execution logic

6. Notification

▪ Notification Service

The NotificationService class integrates Firebase Cloud Messaging with the flutter_local_notifications package.

It creates daily reminders through zonedSchedule() while managing FCM token registration in Firestore for remote notifications.

Each reminder is bound to a unique Android channel to ensure compatibility across OS versions.

The notification logic is fully localized in Arabic and respects user preferences stored in SharedPreferences.

Figure 70-71 shows the scheduling workflow.

Semester-2 1446H (Spring 2025)

```

Future<void> _scheduleDaily(int hour, int minute, int id, String title, String body) async {
    await _localNotifications.zonedSchedule(
        id,
        title,
        body,
        _nextInstanceOfTime(hour, minute),
        const NotificationDetails(
            android: AndroidNotificationDetails(
                'scheduled_channel',
                'Scheduled Reminders',
                importance: Importance.defaultImportance,
                priority: Priority.high,
            ),
            androidScheduleMode: AndroidScheduleMode.exactAllowWhileIdle,
            uiLocalNotificationDateInterpretation: UILocalNotificationDateInterpretation.absoluteTime,
            matchDateTimeComponents: DateTimeComponents.time,
        );
    }

    tz.TZDateTime _nextInstanceOfTime(int hour, int minute) {
        final tz.TZDateTime now = tz.TZDateTime.now(tz.local);
        tz.TZDateTime scheduledDate = tz.TZDateTime(tz.local, now.year, now.month, now.day, hour, minute);
        if (scheduledDate.isBefore(now)) {
            scheduledDate = scheduledDate.add(const Duration(days: 1));
        }
        return scheduledDate;
    }

    Future<void> _saveNotificationToHistory(String? title, String? body) async {
        User? user = _auth.currentUser;
        if (user != null && title != null) {
            await _firestore.collection('users').doc(user.uid).collection('notifications').add({
                'title': title,
                'body': body,
                'timestamp': FieldValue.serverTimestamp(),
                'read': false,
            });
        }
    }
}

```

Figure 48:the scheduling workflow1

```

// --- HELPERS ---

Future<void> saveTokenToDatabase([String? token]) async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    if ((prefs.getBool('notifications_enabled') ?? true) == false) return;

    User? user = _auth.currentUser;
    if (user != null) {
        String? currentToken = token ?? await _fcm.getToken();
        if (currentToken != null) {
            await _firestore.collection('users').doc(user.uid).set({
                'fcmToken': currentToken,
                'lastActive': FieldValue.serverTimestamp(),
            }, SetOptions(merge: true));
        }
    }
}

```

Figure 49:the scheduling workflow2

▪ Notification Settings and History

The `NotificationSettingsPage` provides users with granular control over reminders and push messages.

A switch widget toggles in-app alerts by calling `NotificationService.enableNotifications()` or `disableNotifications()`, persisting the state via `SharedPreferences`.

The `NotificationsPage` retrieves all notifications from `users/{uid}/notifications` and lists them in RTL cards with titles, messages, and timestamps.

A single-action “clear all” button performs a Firestore batch delete for efficient cleanup. Figures 72–75 demonstrate the settings toggle and notification-list rendering.

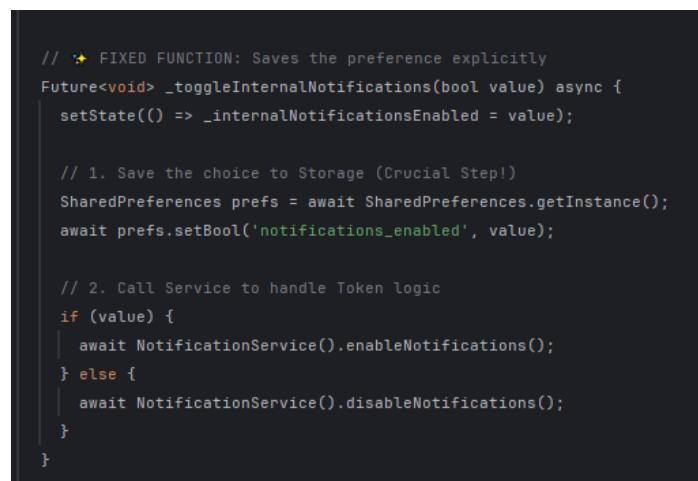


Figure 50: Toggle handler logic

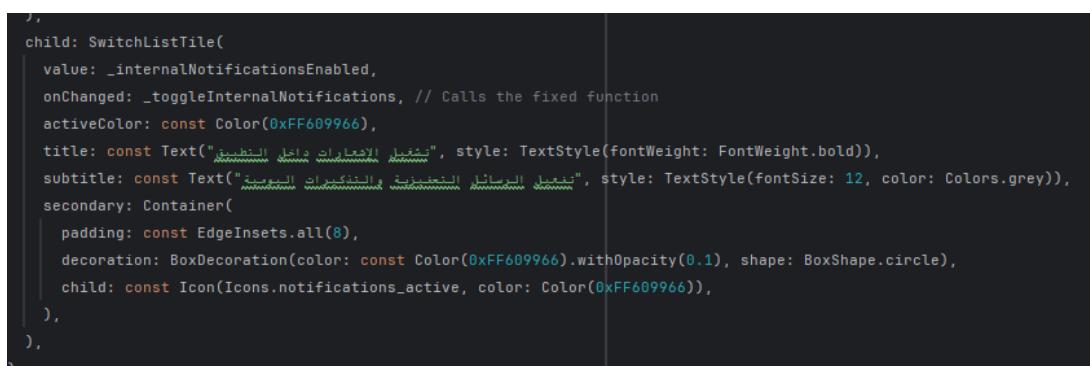


Figure 51: SwitchListTile UI rendering

Semester-2 1446H (Spring 2025)

```
//
body: isLoadingPreference
  ? const Center(child: CircularProgressIndicator(color: Color(0xFF609966)))
  : !_notificationsEnabled
  ? _buildDisabledState()
  : uid == null
  ? const Center(child: Text("الخطابات المنشورة"))
  : StreamBuilder<QuerySnapshot>(
    stream: FirebaseFirestore.instance
      .collection('users')
      .doc(uid)
      .collection('notifications')
      .orderBy('timestamp', descending: true)
      .snapshots(),
    builder: (context, snapshot) {
      if (snapshot.connectionState == ConnectionState.waiting) {
        return const Center(child: CircularProgressIndicator(color: Color(0xFF609966)));
      }

      if (!snapshot.hasData || snapshot.data!.docs.isEmpty) {
        return Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Icon(Icons.notifications_none, size: 80, color: Colors.grey[400]),
              const SizedBox(height: 16),
              Text(
                "لديك خطابات منشورة",
                style: TextStyle(fontSize: 18, color: Colors.grey[600], fontWeight: FontWeight.bold),
              ),
            ],
          );
        );
      }
    });
}

if (!snapshot.hasData || snapshot.data!.docs.isEmpty) {
  return Center(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Icon(Icons.notifications_none, size: 80, color: Colors.grey[400]),
        const SizedBox(height: 16),
        Text(
          "لديك خطابات منشورة",
          style: TextStyle(fontSize: 18, color: Colors.grey[600], fontWeight: FontWeight.bold),
        ),
      ],
    );
}
}
```

Figure 52:Notification list structure and loading state

```
return ListView.builder(
  padding: const EdgeInsets.all(16),
  itemCount: snapshot.data!.docs.length,
  itemBuilder: (context, index) {
    var doc = snapshot.data!.docs[index];
    var data = doc.data() as Map<String, dynamic>;

    String title = data['title'] ?? 'خطاب';
    String body = data['body'] ?? '';
    Timestamp? timestamp = data['timestamp'];

    String timeStr = "بيان";
    if (timestamp != null) {
      try {
        timeStr = DateFormat('h:mm a + d MMM', 'ar').format(timestamp.toDate());
      } catch (e) {
        timeStr = DateFormat('h:mm a + d MMM').format(timestamp.toDate());
      }
    }
  });
}
```

Figure 53:Notification list item rendering

Semester-2 1446H (Spring 2025)

```
Future<void> fetchEducationFavorites() async {
    final userId = FirebaseAuth.instance.currentUser?.uid;
    if (userId == null) return;

    List<Map<String, dynamic>> items = [];

    final lessonsSnap = await FirebaseFirestore.instance
        .collection('Favorites')
        .doc(userId)
        .collection('lessons')
        .get();

    for (var doc in lessonsSnap.docs) {
        final lessonId = doc.id;
        final title = doc['title'];

        // لغات التعليمية التي تم إنشاؤها في الدرس
        final lessonSnapshot = await FirebaseFirestore.instance
            .collection('Learning')
            .doc(lessonId)
            .get();

        final lessonData = lessonSnapshot.data();

        final isSingleVideo = lessonData == null ||
            !lessonData.containsKey('playlist_videos') ||
            (lessonData['playlist_videos'] as List).isEmpty;
    }

    if (isSingleVideo) {
        items.add({'id': lessonId, 'title': title, 'type': 'lesson'});
    }

    final videosSnap = await FirebaseFirestore.instance
        .collection('Favorites')
        .doc(userId)
        .collection('lessons')
        .doc(lessonId)
        .collection('videos')
        .get();

    for (var videoDoc in videosSnap.docs) {
        items.add({
            'id': videoDoc.id,
            'title': videoDoc['subtitle'],
            'url': videoDoc['url'],
            'lessonId': lessonId,
        });
    }
}
```

Semester-2 1446H (Spring 2025)

```
Future<void> removeFavorite(Map<String, dynamic> item) async {
    final userId = FirebaseAuth.instance.currentUser?.uid;
    if (userId == null) return;

    if (item['type'] == 'lesson') {
        await FirebaseFirestore.instance
            .collection('Favorites')
            .doc(userId)
            .collection('lessons')
            .doc(item['id'])
            .delete();
    } else if (item['type'] == 'video') {
        await FirebaseFirestore.instance
            .collection('Favorites')
            .doc(userId)
            .collection('lessons')
            .doc(item['lessonId'])
            .collection('videos')
            .doc(item['id'])
            .delete();
    } else if (item['type'] == 'flashcard') {
        await FirebaseFirestore.instance
            .collection('Favorites')
            .doc(userId)
            .collection('flashcards')
            .doc(item['id'])
            .delete();
    }

    setState(() {
        educationFavorites.removeWhere((element) =>
            element['id'] == item['id'] &&
            element['type'] == item['type'] &&
            (item['type'] != 'video' || element['lessonId'] == item['lessonId']));
    });
}
```

Semester-2 1446H (Spring 2025)

```
class FavoriteLessonIcon extends StatefulWidget {
    final String lessonId;

    const FavoriteLessonIcon({required this.lessonId});

    @override
    _FavoriteLessonIconState createState() => _FavoriteLessonIconState();
}

class _FavoriteLessonIconState extends State<FavoriteLessonIcon> {
    bool isFavorite = false;
    final userId = FirebaseAuth.instance.currentUser?.uid;

    @override
    void initState() {
        super.initState();
        checkIfFavorite();
    }

    Future<void> checkIfFavorite() async {
        if (userId == null) return;

        try {
            final doc = await FirebaseFirestore.instance
                .collection('Favorites')
                .doc(userId)
                .collection('lessons')
                .doc(widget.lessonId)
                .get();

            if (mounted) {
                setState(() {
                    isFavorite = doc.exists;
                });
            }
        } catch (e) {
            debugPrint('Error checking favorite: $e');
        }
    }

    Future<void> toggleFavorite() async {
        if (userId == null) return;

        try {
            final favRef = FirebaseFirestore.instance
                .collection('Favorites')
                .doc(userId)
                .collection('lessons')
                .doc(widget.lessonId);
        }
    }
}
```

Semester-2 1446H (Spring 2025)

```
Future<void> toggleFavorite() async {
    if (userId == null) return;

    try {
        final favRef = FirebaseFirestore.instance
            .collection('Favorites')
            .doc(userId)
            .collection('lessons')
            .doc(widget.lessonId);

        if (isFavorite) {
            await favRef.delete();
        } else {
            final lessonDoc = await FirebaseFirestore.instance
                .collection('Learning')
                .doc(widget.lessonId)
                .get();

            await favRef.set({
                'lessonId': widget.lessonId,
                'title': lessonDoc['title'] ?? '',
                'timestamp': FieldValue.serverTimestamp(),
            });
        }
    }

    if (mounted) {
        setState(() {
            isFavorite = !isFavorite;
        });
    }
} catch (e) {
    debugPrint('Error toggling favorite: $e');
}
}

@Override
Widget build(BuildContext context) {
    return IconButton(
        icon: Icon(
            isFavorite ? Icons.favorite : Icons.favorite_border,
            color: Colors.green[800],
        ), // Icon
        onPressed: toggleFavorite,
    ); // IconButton
}
```

Semester-2 1446H (Spring 2025)

```
class FavoriteVideoIcon extends StatefulWidget {
    final String lessonId;
    final String subtitle;
    final String url;

    const FavoriteVideoIcon({
        required this.lessonId,
        required this.subtitle,
        required this.url,
    });

    @override
    _FavoriteVideoIconState createState() => _FavoriteVideoIconState();
}

class _FavoriteVideoIconState extends State<FavoriteVideoIcon> {
    bool isFavorite = false;
    final userId = FirebaseAuth.instance.currentUser?.uid;

    @override
    void initState() {
        super.initState();
        checkIfFavorite();
    }

    Future<void> checkIfFavorite() async {
        if (userId == null) return;
        final doc = await FirebaseFirestore.instance
            .collection('Favorites')
            .doc(userId)
            .collection('lessons')
            .doc(widget.lessonId)
            .collection('videos')
            .doc(widget.subtitle)
            .get();

        setState(() {
            isFavorite = doc.exists;
        });
    }
}
```

Semester-2 1446H (Spring 2025)

```
Future<void> toggleFavorite() async {
    if (userId == null) return;

    final lessonDocRef = FirebaseFirestore.instance
        .collection('Favorites')
        .doc(userId)
        .collection('lessons')
        .doc(widget.lessonId);

    final videoDocRef =
        lessonDocRef.collection('videos').doc(widget.subtitle);

    if (isFavorite) {
        await videoDocRef.delete();
    } else {
        final lessonExists = await lessonDocRef.get();
        if (!lessonExists.exists) {
            final lessonData = await FirebaseFirestore.instance
                .collection('Learning')
                .doc(widget.lessonId)
                .get();
            final title = lessonData.data()?['title'] ?? '';
            await lessonDocRef.set({'title': title});
        }
    }

    await videoDocRef.set({
        'subtitle': widget.subtitle,
        'url': widget.url,
        'lessonId': widget.lessonId,
    });
}

setState(() {
    isFavorite = !isFavorite;
});

@Override
Widget build(BuildContext context) {
    return IconButton(
        icon: Icon(
            isFavorite ? Icons.favorite : Icons.favorite_border,
            color: Colors.green,
        ), // Icon
        onPressed: toggleFavorite,
    ); // IconButton
}

dependencies:
http: ^1.2.1 # For making REST calls to Gemini
flutter_dotenv: ^5.1.0 # For securely loading GEMINI_API_KEY from a .env file
```

Semester-2 1446H (Spring 2025)

```
Future<void> main() async {
    // Ensure Flutter bindings are initialized (Good!)
    WidgetsFlutterBinding.ensureInitialized();

    // Load Environment Variables for the API Key
    try {

        await dotenv.load(fileName: ".env");
        print("DotEnv loaded successfully.");
    } catch (e) {
        print("Error loading .env file: $e");
    }

    class GeminiService {
        final String? _apiKey = dotenv.env['GEMINI_API_KEY'];
        final String _model = "gemini-1.5-flash";
        final String _apiVersion = "v1beta";

        Future<String?> generateText(String prompt, List<ChatMessage> history) async {
            if (_apiKey == null) return "Error: API Key not configured.";

            final url = Uri.parse(
                "https://generativelanguage.googleapis.com/${_apiVersion}/models/${_model}:generateContent?key=$_apiKey"
            );

            final apiHistory = history.map((msg) => {
                "role": msg.isUserMessage ? "user" : "model",
                "parts": [{"text": msg.text}]
            }).toList();

            final contentPayload = [
                ...apiHistory,
                {"role": "user", "parts": [{"text": prompt}]}
            ];

            final body = json.encode({"contents": contentPayload});

            final response = await http.post(
                url,
                headers: {'Content-Type': 'application/json'},
                body: body,
            );

            if (response.statusCode == 200) {
                final data = json.decode(response.body);
                final candidates = data['candidates'];
                if (candidates != null && candidates.isNotEmpty) {
                    return candidates[0]['content']['parts'][0]['text'];
                }
                return "";
            } else {
                return "Error: ${response.statusCode}";
            }
        }
    }
}
```

Creating natural-sounding educational voiceovers in Arabic was another major challenge.

Semester-2 1446H (Spring 2025)

While many AI voice platforms exist, most lacked strong support for accurate Arabic pronunciation and the Saudi dialect specifically. To address this, we compared multiple AI voice providers, selected the most natural-sounding narrator, and manually added diacritics and pronunciation guides. Finally, we refined the generated output to better align with the Saudi accent, ensuring clarity, cultural relevance, and learner engagement.

1. Prediction model loading delays:

Integrating the AI prediction module introduced noticeable loading delays, especially when fetching both historical prices and prediction documents simultaneously. Because each company required multiple Firestore reads, users occasionally experienced slow initialization and longer waiting times. To address this, we implemented staged loading through the `_bootstrapPage()` method, added rotating status messages to improve perceived responsiveness, and introduced caching to minimize repeated data retrieval.

2. Slow loading of market charts and top-performer data:

Rendering the Market Overview Chart, Company Chart, and the Top Five weekly performers introduced noticeable delays due to the large volume of historical price data required for each component. Fetching multiple Firestore documents, computing weekly changes, and rebuilding chart series—especially when several companies were selected—led to slower response times on mid-range devices. To address this, we implemented data caching, optimized Firestore queries, and added Firestore composite indexes to speed up repeated lookups. These improvements significantly reduced loading times and provided a smoother user experience.

3. Unifying data consistency across all market modules:

One of the major challenges during development was ensuring that all market-related components—such as the Market Overview Chart, Company Chart, Top Five performers, Prediction Model, and Portfolio calculations—used consistent and synchronized data. Because each module fetched historical prices independently and applied different date mappings, slight mismatches sometimes appeared in values and timestamps. To resolve this, we standardized the date-conversion process, unified the Firestore access layer, and introduced shared caching so all components would reference the same data sources. This alignment improved accuracy, eliminated discrepancies, and ensured a coherent user experience across the entire simulation system.

4. Ensuring reliable delivery and control of external notifications:

A key challenge we encountered was managing external push notifications delivered through Firebase Cloud Messaging (FCM) and ensuring that they consistently reached the user. Since FCM delivery can be influenced by device network conditions and Android background restrictions, guaranteeing timely and reliable delivery required additional validation. Another significant challenge was providing users with clear control over enabling or disabling FCM alerts. To address this, we integrated the application directly with the Android system notification settings, allowing users to grant or revoke notification permissions at the OS level. This required synchronizing our in-app preferences with system-level permissions and verifying that the FCM token remained active. These measures helped ensure that external notifications were both permitted and successfully received by the user.

8 System Evaluation

The purpose of this chapter is to present a comprehensive evaluation of the Nomu application and assess its effectiveness in meeting its usability, functionality, and learning objectives. These evaluations provide a detailed understanding of how users perceive the system, how well the application supports investment learning, and the overall quality of the user experience.

8.1 User Acceptance Testing

To measure the usability and user experience of the Nomu application, we employed the System Usability Scale (SUS), a widely used and validated tool for evaluating perceived usability [30]. In addition to SUS, a comprehensive survey based on Nomu's educational and functional objectives was conducted to assess user satisfaction with the core features of Release 2. The survey utilized a five-point Likert scale, where participants indicated their level of agreement from 1 (strongly disagree) to 5 (strongly agree).

Participants were first asked to use the Nomu application exploring lessons, flashcards, simulations, and prediction features and then respond to the survey. A total of 20 responses were collected electronically from individuals who interacted directly with the system. The findings of the SUS evaluation and the objectives-based survey are presented in the following sections.

8.2 Demographics of Participants

The demographics of the users who participated in the UAT are represented in Table 6

Table 6: Demographics of Participants

Variable	Value	Number of participants (Total is 20)
Gander	Male	5
	Female	15
Age	Less than 18	2
	18-24	14
	25-34	1
	35-44	0
	More than 45	3
Technical background	High	5
	Medium	12
	Low	3

8.3 Questionnaire Results

Following each test session, we asked participants several questions to learn more about their experience. Therefore, we have two surveys: System Usability Scale (SUS) and Nomu objective surveys. This section discusses the questionnaire's results.

8.3.1 System Usability Scale (SUS) Results

The System Usability Scale (SUS) was used to evaluate the overall usability and ease of use of the Nomu application. As shown in Table 7, the SUS questionnaire consists of ten standardized statements, where odd-numbered questions represent positive usability aspects (e.g., “I thought the system was easy to use”), while even-numbered questions represent negative usability aspects (e.g., “I found the system unnecessarily complex”). User responses to each question were collected using a five-point Likert scale ranging from Strongly Disagree (1) to Strongly Agree (5), and the distribution of participants’ answers for each question is presented in Table 7.

To calculate the SUS score for each participant, the standard SUS scoring procedure was applied. For every odd-numbered question, the score contribution is calculated as (response value – 1), while for every even-numbered question, the contribution is (5 – response value). The SUS score for each participant is then computed using the formula:

$$\text{SUS Score} = (X + Y) \times 2.5$$

where X is the sum of adjusted scores for all odd-numbered questions, and Y is the sum of adjusted scores for all even-numbered questions. This formula yields a total score ranging from 0 to 100, where higher scores indicate better usability and a more positive user experience.

The individual SUS scores for all 20 participants are summarized in Table 8. Each participant’s SUS score was further mapped to a letter grade and an adjective rating using the standard interpretation guidelines shown in Table 9. According to these guidelines, scores above 80.3 correspond to an “A – Excellent” usability level, scores between 68 and 80.3 fall under “B – Good”, and lower scores reflect decreasing levels of usability.

Overall, the Nomu application achieved an average SUS score of 86.25, which corresponds to a Grade A (Excellent) rating. This indicates a high level of user satisfaction, strong ease of use, and an effective user interface design. The majority of participants assigned very high usability scores, demonstrating that the application is intuitive, easy to navigate, and supportive of users’ learning and simulation activities.

Semester-2 1446H (Spring 2025)

Table 7: SUS Questions and Results

Question	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1. I think that I would like to use this system frequently.	0	0	0	4	16
2. I found the system unnecessarily complex.	16	1	0	0	3
3. I thought the system was easy to use.	0	0	0	1	19
4. I think that I would need the support of a technical person to be able to use this system.	12	2	4	2	0
5. I found the various functions in this system was well integrated.	0	0	0	3	17
6. I thought there was too much inconsistency in this system	14	4	0	1	1
7. I would imagine that most people would learn to use this system very quickly.	0	0	1	4	15
8. I found the system very awkward to use	15	4	0	0	1
9. I felt very confident using this system.	0	1	0	5	14
10. I needed to learn a lot of things before I could get going with this system.	11	4	4	1	0

Table 8: Participant's SUS scores and their interpretation

User	SUS	Grade	Rating
1	60	C	Okay
2	85	A	Excellent
3	70	B	Good
4	60	C	Okay
5	100	A	Excellent

Semester-2 1446H (Spring 2025)

6	82.5	A	Excellent
7	90	A	Excellent
8	70	B	Good
9	95	A	Excellent
10	97.5	A	Excellent
11	100	A	Excellent
12	100	A	Excellent
13	100	A	Excellent
14	100	A	Excellent
15	100	A	Excellent
16	100	A	Excellent
17	100	A	Excellent
18	100	A	Excellent
19	87.5	A	Excellent
20	90	A	Excellent
Average	86.25	A	Excellent

Table 9: Interpretation of SUS score

SUS Score	Grade	Adjective Rating
>80.3	A	Excellent
68-80.3	B	Good
68	C	Okay
51-68	D	Poor
<51	F	Awful

8.3.2 Nomu Objectives Results

A comprehensive user experience survey was conducted to evaluate Nomu's effectiveness in achieving its educational and functional objectives. The survey included 20 participants, most of whom were female university-age users (18–24 years old), with a small number of younger and older

Semester-2 1446H (Spring 2025)

participants, as summarized in Table 6. Participants had varying levels of technical background, though the majority reported a medium level of technical familiarity. After interacting with the application's learning tools, simulation features, and user interface, they responded to questions assessing clarity, engagement, usefulness, and ease of use across Nomu's core functionalities. The results of this survey provide important insights into how well Nomu supports users in understanding investment concepts and navigating the stock market in an accessible and engaging manner.

Overall, feedback toward Nomu's educational components including the daily challenge, educational videos, and interactive flashcards was highly positive. All participants indicated that these tools helped them understand investment concepts more clearly and in a more engaging way. The gamification system also received strong approval, with 45% strongly agreeing and 30% agreeing that the coin-based reward mechanism encouraged them to continue learning and participating in the platform's activities. These findings highlight Nomu's success in motivating learners and promoting consistent interaction with educational content.

The stock market simulation received similarly strong results. A significant 75% strongly agreed that the simulation provided a realistic and clear understanding of market behavior, with an additional 20% agreeing. Participants also demonstrated high satisfaction with the market overview page, 60% strongly agreed and 40% agreed that the clear presentation of profits, losses, and market indicators helped them better understand their financial performance. Likewise, 55% strongly agreed and 45% agreed that the charts and visual tools enhanced their understanding of stock movement and price fluctuations. These results demonstrate that Nomu effectively simplifies complex financial data and builds users' confidence in exploring investment scenarios.

Participants also expressed high satisfaction with Nomu's user interface and navigation experience. An outstanding 95% strongly agreed that the interface is simple, clean, and easy to use. Additionally, 80% strongly agreed and 15% agreed that navigation across the application was smooth and intuitive, confirming that users encountered minimal friction overall. The overall perception of usability is very positive, with participants emphasizing that the application's structure helped them understand investment activities more easily.

The survey also showed that Nomu significantly improved users' confidence and understanding of investment concepts. A total of 65% strongly agreed and 35% agreed that the application enhanced their practical knowledge of investment and trading. Furthermore, 90% strongly agreed that Nomu increased their awareness and understanding of investment practices in general, indicating that the application effectively supports beginner-level users and contributes to financial literacy development. Participants also expressed strong approval of the favorite's features, with 75% strongly agreeing that these tools helped them organize their learning and access important content more efficiently.

Semester-2 1446H (Spring 2025)

Participants' overall impressions of Nomu were overwhelmingly positive. Many described the application as smooth, beginner-friendly, well-designed, and highly useful for understanding stock market fundamentals. Several users highlighted that Nomu provides a unique educational experience in Arabic, filling a gap left by foreign investment applications and presenting financial concepts in a simplified and accessible manner. Overall, the survey reveals that Nomu offers a motivating, intuitive, and engaging learning environment for users.

Participants also provided valuable suggestions for future improvement, including adding an English version, supporting multiple languages, expanding the number of available companies, integrating real-time data, and offering live learning sessions with experts. These suggestions will help guide the next phase of Nomu's development.

In summary, the user experience survey demonstrates that Nomu successfully simplifies investment concepts, enhances learning, and provides a highly intuitive interface. The strong levels of user satisfaction indicate that Nomu effectively supports beginners in building financial knowledge and engaging confidently with stock market activities. The detailed objectives evaluation results of this survey are summarized in Table 20 in Appendix E.

8.4 Quality Attributes (NFR testing)

Non-functional testing was conducted to ensure that Nomu meets the key quality attributes defined for the system, namely performance, usability, learnability, and reliability/feedback. Table 10 summarizes the NFR test scenarios derived from selected user stories, the measures and acceptance criteria used, and the results from 20 participants. Performance was evaluated using screen loading times under normal network conditions, usability through icon and label recognizability, learnability by measuring how quickly new users could complete core tasks, and reliability/feedback by observing how fast and clearly validation error messages appeared. Overall, Nomu met the defined thresholds for performance, learnability, and feedback responsiveness, while usability testing revealed that some icons and labels, particularly on the Portfolio Page, still require refinement to improve immediate recognizability.

Semester-2 1446H (Spring 2025)

Table 10: NFR Testing Results

User Story	Quality Attribute	Measure + Acceptance Criteria	Results (including test steps & 20 users)
As a user, I want the application's screens to load within 10 seconds under normal network conditions.	Performance How fast do Nomu's main screens load under normal network conditions?	<ul style="list-style-type: none"> Measure loading time for Dashboard, Learning, Simulation, and Predictions pages. Acceptance Criteria: All pages must load in less than 10 seconds with smooth transitions. 	<p>Test Scenario:</p> <ol style="list-style-type: none"> User launches the app. Navigates between home page, Learning, Simulation and Predictions pages. Loading times measured for each page. <p>Participants: 20 users.</p> <p>Results:</p> <ul style="list-style-type: none"> Loading times: Min 1.2s, Max ~10s, Avg 6.3s. Simulation Page reached ~10s but met requirement. Performance requirement met.
As a user, I want app icons and labels to be recognizable within 5 seconds.	Usability How easily and quickly can users understand the meaning of Nomu's icons and labels within 5 seconds?	<ul style="list-style-type: none"> Measure icon/label recognition time. Acceptance Criteria: $\geq 90\%$ of users recognize icons in ≤ 5 seconds. 	<p>Test Scenario:</p> <ol style="list-style-type: none"> Users shown main navigation icons and labels. Asked to identify each icon's and label meaning within 5 seconds. <p>Participants: 20 users.</p> <p>Results:</p> <p>Only 13 out of 20 users recognized icons and labels within 5 seconds. Most difficulties came specifically from the labels displayed on the Portfolio Page, where users needed extra time to understand their meaning.</p> <p>Requirement partially met</p>
As a user, I want the system to be easy to learn and usable within 20–30 minutes.	Learnability How quickly can new users understand and use Nomu within 20–30 minutes of first use?	<ul style="list-style-type: none"> Users should complete tasks in ≤ 30 mins. 	<p>Test Scenario:</p> <ol style="list-style-type: none"> users perform key tasks: navigation, watching lesson, using simulation, checking predictions. Time and errors recorded.

Semester-2 1446H (Spring 2025)

			<p>Participants: 20 users.</p> <p>Results:</p> <ul style="list-style-type: none"> • 19/20 users completed tasks within the required time. • Completion times: 12–32 mins, Avg 22 mins. <p>Learnability requirement met.</p>
As a user, I want error messages to appear within 5 seconds after invalid input.	<p>Reliability / Feedback</p> <p>How quickly does Nomu display validation error messages after the user submits invalid input?</p>	<ul style="list-style-type: none"> • Error messages should appear in ≤ 5 seconds. • Messages must provide corrective guidance. 	<p>Test Scenario:</p> <ol style="list-style-type: none"> 1. Users intentionally entered invalid inputs (wrong login, empty form, invalid data). 2. Response time measured. <p>Participants: 20 users.</p> <p>Results:</p> <ul style="list-style-type: none"> • Error messages displayed in 0.5–1.2 seconds. <p>Feedback requirement fully met.</p>

8.5 Discussion

The system evaluation results indicate that the Nomu application largely meets its intended goals in terms of usability, learnability, and supporting beginners in investment learning. The SUS average score of 86.25 (Grade A – Excellent) shows that users found the system easy to use, well-integrated, and that most people could learn it quickly. The objectives-based questionnaire supports this, as participants strongly agreed that the learning tools, simulation, portfolio, and charts helped them understand investment concepts in a simple and engaging way.

The non-functional requirements (NFRs) were also mostly satisfied. Screen loading times stayed within the 10-second limit under normal network conditions, and new users were able to learn and use the system within 20–30 minutes, with a low error rate. Error messages appeared quickly (within about 1 second), providing clear feedback. However, one requirement was only partially met: only 13 out of 20 users recognized icons and labels within 5 seconds, especially on the Portfolio page. This suggests that while the overall design is usable, some labels and navigation elements need clearer wording or visual cues.

Overall, the results are good and aligned with the project objectives, but improvement is still possible. Future work can focus on refining icons and labels (especially in the portfolio and trading sections), slightly optimizing performance on data-heavy pages like Simulation and Market Prediction, and iterating the design based on further user testing to remove the remaining points of confusion.



07 System Testing



9 System Testing

In this section, we present Nomu's system testing starting with User Acceptance Testing via structured surveys and interviews, then cover the methodology, participant demographics, key findings, and recommendations for future enhancements.

9.4 User Acceptance Testing

User Acceptance Testing (UAT) is a crucial phase in the evaluation of the Nomu application, aimed at determining whether the system meets the expectations and needs of end users from the intended target audience. UAT was conducted with a total of 15 participants: 10 individuals completed a structured online survey, and 5 participants took part in in-depth interviews. This dual approach enabled the collection of both quantitative data through questionnaires and qualitative insights through personal experiences and reflections. The survey responses were structured around the application's key non-functional requirements, including usability, ease of navigation, visual clarity, and overall satisfaction, while the interviews further examined these aspects and gathered deeper user insights regarding the application's strengths, limitations, and suggestions for improvement.

9.4.1 Testing Design and Methodology

The UAT process involved selecting participants based on their alignment with the application's intended user base, primarily individuals with limited financial literacy or beginner-level investment knowledge. This ensured that the system would be evaluated by users who reflect the core audience that Nomu aims to support. Participants were asked to explore the application's main features, including the interfaces, navigation structure, flashcard-based learning system, and the daily investment challenge.

The structured survey questionnaire included both demographic and functional questions to assess the system's performance from a user standpoint. Responses were collected via Google Forms and later analyzed in Google Sheets, with visual aids such as charts and tables generated to support the data analysis.

To complement the survey, semi-structured interviews were conducted with five participants. These interviews provided more nuanced insights into user experiences, including emotional engagement, ease of understanding, and suggestions for future improvement. Participants reflected on both the strengths and potential enhancements of the application, enriching the evaluation with authentic user perspectives.

9.4.2 Demographics of Participants

To ensure clarity and accuracy, the demographics of participants are separated into two groups: survey respondents and interview participants. The survey demographics are supported by visual charts, while the interview data is summarized narratively, as it was collected in a qualitative format.

Survey Participants (n = 10):

Survey respondents represented a varied group that broadly reflects the application's intended user base. Their demographic characteristics are illustrated in Figures 62 to 65.

- Age Distribution: 50% of participants were aged 18–25, 30% were between 26–30, 10% were between 31–45, and one participant (10%) was in the 46–65 age group (Figure 62).
- Gender: The group included 70% females and 30% males (Figure 63).
- Education Level: 90% of the participants held or were pursuing a bachelor's degree, while 10% had completed high school (Figure 64).
- Experience with Investment Applications: 30% of the users had previous experience with financial or investment-related applications, while 70% did not (Figure 65).

Interview Participants (n = 5):

The interview group consisted of a diverse set of users that added depth and qualitative context to the evaluation. Among the five interviewees:

- Four were university students, all identifying as beginners in investment knowledge.
- One participant was a 50-year-old employee, bringing a more experienced and professional perspective to the feedback.
- The group included both male and female participants, aligning with the survey's gender balance.
- All interviewees identified themselves as beginners in financial knowledge, except for one participant who considered themselves at an intermediate level, reinforcing the system's relevance to its intended audience.

Semester-2 1446H (Spring 2025)

What is your age?

- 46–65
- 18–25
- 26–30
- 31–45



Figure 54 Age distribution

What is your gender?

- Female
- Male

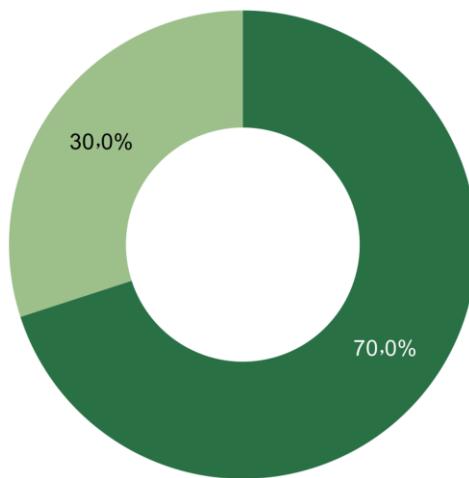


Figure 55:Gender distribution

Semester-2 1446H (Spring 2025)

What is your highest level of education completed?

- Bachelor's Degree
- High School

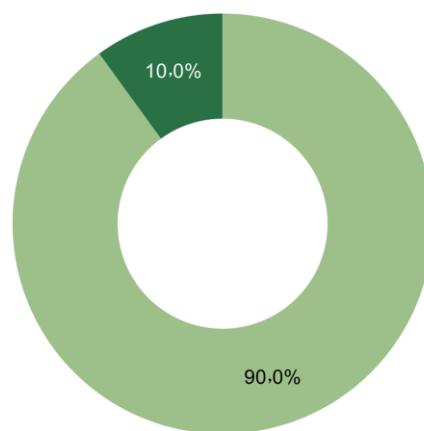


Figure 56: Education Level

Have you used investment or finance-related apps before?

- No
- Yes

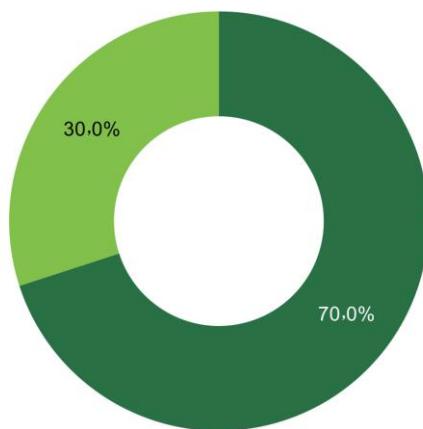


Figure 57: Experience with Investment Applications

9.4.3 Questionnaire/Interview Results

This section presents the findings from the User Acceptance Testing (UAT), which included two evaluation components: a structured questionnaire completed by 10 participants and semi-structured interviews conducted with 5 individuals. The questionnaire results are visualized in charts (see Appendix C), while the interview responses are included in (Appendix D).

Survey Results

- Participants rated the application using a **5-point Likert scale** (1 = Strongly Disagree, 5 = Strongly Agree). The aggregated results are summarized below and supported by visual charts in Appendix C.
- **Navigation and Usability:** All survey participants (100%) strongly agreed that the application was easy to navigate, praising its layout and simplicity.
- **Error Handling and Clarity:** The majority rated the application positively for providing clear feedback during user errors or incorrect inputs.
- **Learning Tools (Flashcards & Daily challenges):** Educational tools were well-received, with 50% agreeing, 40% strongly agreeing, and 10% neutral about their effectiveness in enhancing financial knowledge (see Figure 66).
- **Ease Learning:** 60% strongly agreed and 40% agreed that they could quickly understand and use the system with little to no guidance (Figure 67).
- **Recognizability of Icons:** 80% strongly agreed and 20% agreed that application icons and visual labels were clear and easy to understand (Figure 68).
- **Willingness to Use the System:** Participants expressed unanimous satisfaction, with all users strongly agreeing that they would like to use the Nomu application frequently (Figure 69). This endorsement held true across a diverse cohort in terms of age, gender, and prior experience, confirming that the application is both intuitive and easy to use—even participants new to financial tools found the system straightforward to learn and operate.

Semester-2 1446H (Spring 2025)

The daily questions and flashcards improved my financial knowledge

- 3
- 4
- 5

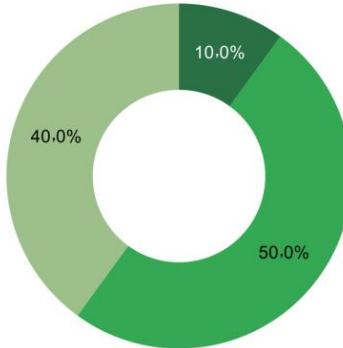


Figure 58 Learning Tools (Flashcards & Daily challenges)

I was able to understand and use most features of the app without needing much time or external help

- 4
- 5

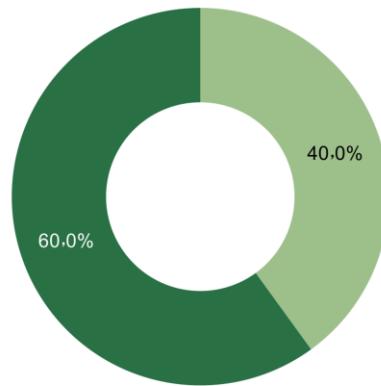


Figure 59 Ease Learning

I was able to recognize icons and labels within a few seconds without confusion

- 4
- 5

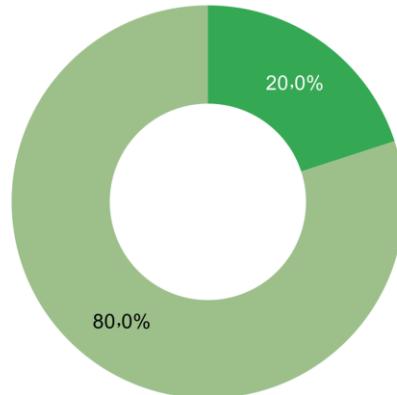


Figure 60 Recognizability of Icons

I think I would like to use this system frequently

- 5

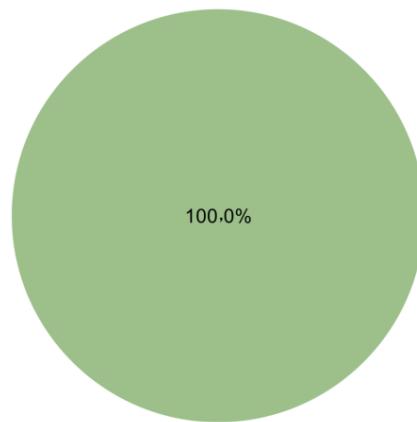


Figure 61 Willingness to Use the System

Interview Results

The five interviews offered deeper insights into the user experience that cannot be captured through numeric ratings alone. Most interviewees were university students with beginner-level investment knowledge, while one participant was a 50-year-old employee, providing a more experienced and professional perspective. Thematic findings include:

- **Clarity and Simplicity:** All interviewees emphasized that the application was easy to use, frequently describing the interface and navigation as clear and straightforward.
- **Educational Value:** Participants appreciated the structured and interactive nature of the learning tools, describing the flashcards and daily questions as helpful and well-organized.
- **Gamification Feedback:** The coin reward system received positive feedback, with users noting that it increased their motivation to engage with the content, even though it was not their sole reason for participation.
- **Trust and Recommendation:** All interviewees stated that they would recommend the application to others, expressing strong confidence and satisfaction in their responses.
- **Improvement Suggestions:** Interviewees proposed enhancements such as supporting multiple languages and introducing networking features to connect with peers or financial mentors.

9.5 Discussion

Overall, the evaluation revealed that the Nomu application performed strongly in key areas of usability and educational value. Participants consistently reported that the application was intuitive and easy to navigate, with survey responses reflecting unanimous agreement on its clear structure. Interviewees reinforced this finding, describing the user interface as well-organized and the learning materials as engaging and easy to follow. The inclusion of various learning tools, such as educational videos, interactive flashcards, and daily challenges, was especially well-received. Users emphasized the value of these components in helping them understand investment concepts in a simplified and structured way. Additionally, the coin-based gamification system was noted as a motivating factor, encouraging users to continue engaging with the learning material. Most importantly, nearly all participants expressed trust in the content and a high likelihood of recommending the application to others.

Semester-2 1446H (Spring 2025)

Despite these strengths, some participants provided constructive suggestions to further improve the application. One common recommendation was the addition of multilingual support, which would enhance accessibility for users with different language preferences and expand the platform's reach. Others recommended introducing more interactive features, such as a space to interact with peers and experts. These enhancements were seen as valuable additions that could boost engagement and provide a more personalized user experience.

In conclusion, the findings indicate that Nomu is a highly usable and educationally effective platform tailored to novice investors. It successfully integrates an intuitive user interface, engaging content, and motivational elements to support learning. Moving forward, incorporating user suggestions, particularly those related to language support and interactivity, will further strengthen the platform's impact and scalability. These improvements will ensure that Nomu continues to meet the evolving needs of its users while maintaining its educational mission.



08 Conclusions and Future Work



10 Conclusions and Future Work

10.4 Conclusions

The Nomu application emerges as a robust and innovative solution designed specifically to address the widespread challenge of financial illiteracy and investment hesitancy, particularly prevalent among beginner investors and individuals with limited financial resources. Through its thoughtfully designed educational approach, practical simulations, and gamification strategies, Nomu creates an engaging, risk-free environment for learning essential investment skills. The development of this application not only contributes substantially to personal financial empowerment but also supports broader economic objectives such as financial stability, economic diversification, and increased market participation, particularly aligning with Saudi Arabia's Vision 2030 [31].

Nomu successfully bridges the gap between theoretical financial concepts and practical investment skills by incorporating advanced technological tools such as machine learning algorithms, interactive user interfaces, and gamified features. These elements collectively create an immersive learning experience, significantly enhancing user engagement and practical understanding of investment strategies. Despite encountering several technical and methodological challenges during development, the team effectively navigated these obstacles through diligent research, agile methodologies, and collaborative teamwork.

10.5 Global and Local Impact

While Nomu is primarily designed as an Arabic-language application catering to Arabic-speaking users, its approach has significant global implications. The methodology and technologies employed by Nomu, such as gamified learning, realistic market simulations, and machine learning predictive capabilities, are universally applicable. Thus, Nomu serves as a scalable model for financial education applications globally, demonstrating effective strategies that other countries and communities could adopt. By showcasing an innovative integration of educational content, practical simulations, and technological enhancements, Nomu potentially inspires similar initiatives around the world, contributing to global efforts to improve financial literacy and investment readiness.

Moreover, as Arabic is a widely spoken language internationally, including countries beyond Saudi Arabia such as Egypt, the UAE, Morocco, and others, Nomu's impact extends regionally and internationally within Arabic-speaking communities. The principles, strategies, and successes derived from Nomu's implementation could be adapted into various linguistic and cultural contexts, amplifying its potential global impact.

Locally, particularly in Saudi Arabia, the impact of Nomu is especially pronounced. The application directly supports Saudi Vision 2030 by empowering citizens with financial literacy and investment confidence necessary for active participation in the economy. Saudi Arabia's vision of reducing economic dependence on oil through diversification is supported by Nomu's educational initiatives, fostering an environment where individuals are encouraged to explore investment opportunities. Additionally, the Arabic-language focus significantly reduces linguistic and cultural barriers, promoting inclusivity and accessibility within the region [29].

10.6 Problems and Challenges Encountered During Software Development

During the development of the Nomu application, our team encountered several technical and organizational challenges that strengthened our adaptability, teamwork, and technical skills. Below are the main issues we faced and how we resolved them:

1. Voice Generation for Educational Videos:

Creating natural-sounding voiceovers in the Saudi Arabic dialect was difficult using AI tools. After testing various platforms, we subscribed to a paid service that delivered high-quality, culturally appropriate audio.

2. Combining Application Pages:

Each team member initially worked on separate pages, leading to inconsistencies. We resolved this through regular meetings and coordinated code integration.

3. Gathering Stock Data:

We needed diverse stock data (rising, falling, stable) to build realistic simulations. This was achieved using official data purchased directly from the Tadawul website.

4. Learning Flutter:

As Flutter was new to us, we faced limitations in customization. YouTube tutorials helped us learn how to effectively use the platform, as well as asking ChatGPT for guidance whenever we encounter challenges.

5. Firebase Authentication Customization:

We replaced Firebase's default messages and flows with customized ones to improve clarity and user experience.

6. Reset Password Link Customization:

Adjusting Firebase's reset password link to match our security standards was challenging and required deep configuration of dynamic links.

These challenges provided valuable learning opportunities and directly contributed to building a well-structured and user-friendly final product.

10.7 Limitations of the System

While the Nomu application introduces an innovative and user-centered approach to investment education, it still faces several limitations that need to be acknowledged and addressed in future versions. Below is a summary of the key limitations identified:

1. Reliance on Historical Data:

Nomu currently uses historical market data to simulate investment scenarios. While this approach provides a safe and risk-free environment for users to practice strategies, it does not reflect real-time market dynamics or sudden economic changes. As a result, users may not fully experience the unpredictability or volatility of actual financial markets, which could reduce the educational value in highly dynamic market situations.

2. Arabic-Language Only Interface:

The application currently supports only the Arabic language, which makes it highly accessible to its primary audience in Saudi Arabia and other Arabic-speaking regions. However, this language limitation prevents users from other regions or non-Arabic speakers from using the application, limiting its potential global reach and scalability.

3. Requirement for Constant Internet Connection:

Many core features of Nomu, including loading lessons, retrieving stock data, and running simulations, require a stable internet connection. Users in areas with poor connectivity may experience interruptions or be unable to access important parts of the application. This limits its usability in remote or low-bandwidth regions and may affect user retention.

4. Limited Platform Support:

Currently, the application only supports Android devices. This limitation excludes iOS users and reduces the overall accessibility of the application across diverse user bases.

By recognizing these limitations, future updates to Nomu can be better aligned with user needs, ensuring improved scalability, accessibility, and realism in investment simulation and education.

10.8 Main Contributions of the Nomu

Nomu's main contribution lies in the development of an interactive, Arabic-language investment education platform that empowers beginner users to build financial knowledge and practice investment skills in a safe, engaging environment. The application combines simulation, education, and prediction to simplify stock market learning and promote financial inclusion.

1. Realistic Stock Market Simulations:

Nomu enables users to engage with realistic stock market scenarios using historical data. Through virtual portfolios, users can test investment strategies, monitor performance, and develop hands-on experience without financial risk. This provides a foundation for real-world readiness and confidence building in a low-pressure environment.

2. Integration of Predictive Models for Learning:

The application uses machine learning techniques—such as ARIMA and classification algorithms—to predict short-term price trends. These models allow users to learn how data-driven analysis supports investment decisions, introducing them to modern financial tools in a practical and simplified format.

3. Arabic-Language Educational Environment:

One of Nomu's core contributions is its cultural and linguistic accessibility. By offering a fully Arabic interface and localized content, the application breaks down common language and cultural barriers in financial education, especially for users in Saudi Arabia and the wider Arab region.

4. Gamified Learning to Boost Engagement:

Nomu incorporates gamification features like daily quizzes, coin rewards, and challenge-based learning to motivate consistent user interaction. These features make complex financial concepts more enjoyable and memorable, increasing user engagement and retention over time.

5. Alignment with Saudi Vision 2030 Objectives:

The application directly supports the financial literacy goals outlined in Saudi Arabia's Vision 2030. By encouraging citizens to understand investment principles and engage with the stock market, Nomu promotes economic participation and contributes to long-term national goals related to economic diversification and digital transformation.

In summary, Nomu delivers a scalable and locally adapted solution for investment education. Its combination of practical simulations, predictive technology, gamification, and cultural relevance makes it a benchmark tool for financial literacy in the Arabic-speaking world.

10.9 Future work

While the initial version of the Nomu application has achieved notable success in providing an engaging and beginner-friendly platform for investment education, there remain several promising opportunities for enhancement and expansion. The following future developments are proposed to improve the application's effectiveness, scalability, and overall user experience:

1. Real-Time Data Integration:

Future updates will focus on integrating live stock market data, allowing users to interact with actual market movements. This feature will enhance the realism of the simulation environment and help users develop better timing and decision-making skills in dynamic market conditions.

2. Multi-Language Support:

To broaden accessibility, the application is planned to support additional languages, including English. This enhancement will enable Nomu to reach a more diverse user base and be adopted in multilingual educational and training environments.

3. Advanced Gamification Features:

Enhancing the gamification system is a key priority. Future versions will introduce more engaging features such as experience levels, achievement badges, leaderboards, and interactive user challenges to increase user motivation, retention, and long-term learning commitment.

4. Personalized Learning Paths:

Nomu will offer tailored learning paths based on users' prior knowledge and skill levels (beginner, intermediate, or advanced). This approach ensures that users receive content that is appropriate for their understanding, encouraging gradual and effective learning progression.

5. Multi-Platform Compatibility

To ensure broader reach and accessibility, Nomu will expand to support multiple platforms beyond mobile devices. The development roadmap includes launching web and tablet versions of the application, making it easier for users to learn and invest using their preferred devices—whether they are at home, in a classroom, or on the go.

By implementing these enhancements, Nomu aims to evolve into a more comprehensive, adaptive, and user-centered investment education platform—equipped to serve a wide range of learners and empower them to make informed financial decisions.

11 Acknowledgements

First and foremost, we would like to thank Allah, the Most Gracious and Most Merciful, for granting us the strength, patience, and ability to complete this graduation project. Without His blessings, this work would not have been possible.

We would also like to express our sincere gratitude to our supervisors, Dr. Qatrunnada Alsmail and Dr. Yousra Saud Almathami, for their continuous guidance, support, and valuable feedback throughout all stages of this project. Their insights, encouragement, and dedication greatly contributed to shaping the Nomu application and this report.

Our appreciation also extends to the faculty and staff of the Information Technology Department at King Saud University for providing us with the knowledge, skills, and academic environment that enabled us to successfully complete this work.

We are deeply grateful to our families for their constant support, prayers, and understanding. Their encouragement and belief in us have been a major source of motivation throughout our study journey.

Finally, we would like to thank our friends, classmates, and all participants who took part in the User Acceptance Testing of Nomu. Their time, feedback, and suggestions were invaluable in improving the quality of the application and this project.

To everyone who contributed in any way, we sincerely thank you.



09 References



12 References

- [1] Vision 2030, "Public Investment Fund (PIF) Program," [Online]. Available: <https://www.vision2030.gov.sa/en/explore/programs/public-investment-fund-program>. [Accessed: Jan. 13, 2025].
- [2] Tamra Capital, "Tamra Application," [Online]. Available: <https://tamracapital.sa/>. [Accessed: Jan. 18, 2025].
- [3] Saudi Central Bank, "Measuring Financial Capability: The context of Saudi Arabia," [Online]. Available: <https://sama.gov.sa/en-US/EconomicResearch/WorkingPapers/Measuring%20Financial%20Capability%20%20The%20context%20of%20Saudi%20Arabia.pdf>. [Accessed: Jan. 13, 2025].
- [4] Atlassian, "Jira Software | Issue & Project Tracking," *Atlassian*, [Online]. Available: <https://www.atlassian.com/software/jira>. [Accessed: Dec. 06, 2025].
- [5] GitHub, "GitHub: Let's build from here," *GitHub*, [Online]. Available: <https://github.com/>. [Accessed: Dec. 06, 2025].
- [6] Flutter, "Build apps for any screen," *Flutter.dev*, [Online]. Available: <https://flutter.dev/>. [Accessed: Dec. 06, 2025].
- [7] Google, "Firebase Documentation," *Firebase*, [Online]. Available: <https://firebase.google.com/>. [Accessed: Dec. 06, 2025].
- [8] Canva, "Free Design Tool: Presentations, Video, Social Media," *Canva*, [Online]. Available: <https://www.canva.com/>. [Accessed: Dec. 06, 2025].
- [9] PlayHT, "AI Voice Generator - Realistic Text to Speech," *Play.ht*, [Online]. Available: <https://play.ht/>. [Accessed: Dec. 06, 2025].
- [10] Saudi Exchange, "Saudi Exchange (Tadawul)," *SaudiExchange.sa*, [Online]. Available: <https://www.saudiexchange.sa/>. [Accessed: Dec. 06, 2025].
- [11] Python Software Foundation, "Python," *Python.org*, [Online]. Available: <https://www.python.org/>. [Accessed: Dec. 06, 2025].
- [12] Massachusetts Institute of Technology, "Study on Financial Literacy and Investment Knowledge," 2019, [Online]. Available: <https://sloan.mit.edu/>. [Accessed: Dec. 06, 2025].
- [13] J. Jankowski and T. Shank, "A Comparison of Online Stock Trading Simulators for Teaching Investments," *Journal of Financial Education*, vol. 36, no. 1/2, pp. 105-119, 2010, [Online]. Available: https://digitalcommons.usf.edu/fac_publications/785/. [Accessed: Dec. 06, 2025].
- [14] S. Dressler, T. Rachfall, A. Kapanen, and D. Foerster-Trallo, "Improved Learning Performance Due to the Implementation of a Stock Market Game – A Case Study About the Possibility of Enhancing Student Skills with Game-Based Learning," in *ICERI2016 Proceedings*, 2016, pp. 2974-

Semester-2 1446H (Spring 2025)

2980, [Online]. Available: <https://library.iated.org/view/DRESSLER2016IMP>. [Accessed: Dec. 06, 2025].

[15] K. Patil, "Impact of Stock Market Simulators on Financial Analytical Skills in Technical Education," *Journal of Emerging Engineering Trends*, vol. 37, no. Special Issue-2, p. 298, 2024, [Online]. Available: <https://journaleet.in/index.php/jeet/article/view/2487>. [Accessed: Dec. 06, 2025].

[16] J. Hamari, J. Koivisto, and H. Sarsa, "Does Gamification Work? – A Literature Review of Empirical Studies on Gamification," in *Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS)*, 2014, pp. 3025-3034, [Online]. Available: <https://ieeexplore.ieee.org/document/6758978>. [Accessed: Dec. 06, 2025].

[17] I. Caponetto, J. Earp, and M. Ott, "Gamification and Education: A Literature Review," in *Proceedings of the European Conference on Games-Based Learning*, vol. 1, p. 50, 2014, [Online]. Available: <https://iris.cnr.it/handle/20.500.14243/226324>. [Accessed: Dec. 06, 2025].

[18] P. Bitrián, I. Buil, and S. Catalán, "Making Finance Fun: The Gamification of Personal Financial Management Apps," *International Journal of Bank Marketing*, vol. 39, no. 7, pp. 1310-1332, 2021, [Online]. Available: <https://doi.org/10.1108/IJBM-02-2021-0074>. [Accessed: Dec. 06, 2025].

[19] A. Pal, K. Indapurkar, and K. P. Gupta, "Gamification of Financial Applications and Financial Behavior of Young Investors," *Young Consumers*, vol. 22, no. 3, pp. 503-519, 2021, [Online]. Available: <https://doi.org/10.1108/YC-10-2020-1240>. [Accessed: Dec. 06, 2025].

[20] Baptista, G., & Oliveira, T. (2017). Why so serious? Gamification impact in the acceptance of mobile banking services. *Internet Research*, 27(1), 118–139. Available: <https://doi.org/10.1108/IntR-10-2015-0296>. [Accessed: Dec. 06, 2025].

[21] Looyestyn, J., Kernot, J., Boshoff, K., Ryan, J., Edney, S., & Maher, C. (2017). The effectiveness of gamification apps for increasing engagement: A systematic review. *PLOS ONE*, 12(3), e0173403. Available: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0173403>. [Accessed: Dec. 06, 2025].

[22] Rodrigues, L. F., Costa, C. J., & Oliveira, A. (2016). Gamification: A framework for designing software in e-banking. *Computers in Human Behavior*, 62, 620–629. Available: <https://doi.org/10.1016/j.chb.2016.03.015>. [Accessed: Dec. 06, 2025].

[23] Mubasher Financial Services, "Tadawul Trading Simulator (MTrade)," *Google Play Store*, [Online]. Available: <https://play.google.com/store/apps/details?id=com.mfs.mtrade.tdwlgme>. [Accessed: Dec. 06, 2025].

[24] Capital.com, "Investmate: Learn to Trade," *Google Play Store*, [Online]. Available: <https://play.google.com/store/apps/details?id=com.capital.investmate>. [Accessed: Dec. 06, 2025].

Semester-2 1446H (Spring 2025)

- [25] Finance Illustrated, "Trading Game: Stocks & Forex," *Google Play Store*, [Online]. Available: <https://play.google.com/store/apps/details?id=com.tiim.tradinggame>. [Accessed: Dec. 06, 2025].
- [26] A-Life Software, "Stock Trainer: Virtual Trading," *Google Play Store*, [Online]. Available: <https://play.google.com/store/apps/details?id=com.alifesoftware.stocktrainer>. [Accessed: Dec. 06, 2025].
- [27] Three Investeers, "Stock Market Simulator: Game," *Google Play Store*, [Online]. Available: <https://play.google.com/store/apps/details?id=com.threeinvesteers.app>. [Accessed: Dec. 06, 2025].
- [28] B. Graham, "The Intelligent Investor," New York, NY, USA: Harper Business, 2006. [Online]. Available: <https://www.goodreads.com/book/show/106835.The+Intelligent+Investor> [Accessed: Apr. 27, 2025].
- [29] Grassi, P. A., Garcia, M. E., & Fenton, J. L. (2017). *Digital Identity Guidelines: Authentication and Lifecycle Management* (NIST Special Publication 800-63B). National Institute of Standards and Technology. Available at: <https://doi.org/10.6028/NIST.SP.800-63b>
- [30] K. Betteridge, "What Every UXC Client Should Know About SUS Scores," Bentley University User Experience Center, 2021. Available at: <https://www.bentley.edu/centers/user-experience-center/what-every-client-should-know-about-sus-scores> (accessed 5 Dec. 2025).
- [31] Saudi Vision 2030, "Vision 2030," [Online]. Available: <https://vision2030.gov.sa>. [Accessed: May 1, 2025].
- [32] ElevenLabs, "ElevenLabs AI Voice Generator & Text to Speech." [Online]. Available: <https://elevenlabs.io/>. [Accessed: Dec. 8, 2025].
- [33] NabaRati, "NabaRati | Arabic Voiceovers, Transcription & Translation." [Online]. Available: <https://www.nabarati.com/>. [Accessed: Dec. 8, 2025].



10 Appendices



13 Appendices

13.4 Appendix A: Interviews

This section presents insights from user interviews conducted to understand their needs and requirements, ensuring the project aligns with their expectations. The questions and answers from the interviews are presented in Table 11 to 14.

Table 11: First Interview Outline

Interview#1	
Interview Date: Jan 22, 2025 Start Time: 7:00 PM End Time: 7:24 PM	
Questions:	Answers:
Q1: هل جربت الاستثمار من قبل؟	A1: لا
Q2: هل لديك دافع لتعلم الاستثمار؟ اذا كان جوابك بنعم ما هي دوافعك؟	A2: نعم، أرغب بتعلم كيفية تنمية اموالي وتطوير مهاراتي المالية على حد سواء، أيضا الاستفادة من الاستثمار على المدى الطويل فانا الذي خطط مستقبلية، زيادة مصادر الدخل
Q3: ما هي الصعوبات التي تواجهك في تعلم الاستثمار؟	A3: من الصعوبات التي واجهتها عدم الفهم العام للاستثمار، وعدم وجود منصة موجهة ومرشدة لل الاستثمار، أيضا الخوف من الخسارة
Q4: كيف تفضلين ان تكون طريقة تعليم الاستثمار؟	A4: خطوات نصية مكتوبة خطوة بخطوة ومقاطع قصيرة
Q5: هل سبق لك تجربة استخدام تطبيقات محاكاة لتعلم الاستثمار؟ ان كانت اجابتك نعم، كيف كانت تجربتك؟	A5: لا، لم يسبق لي بتجربتها
Q6: هل وجود الألعاب والمحفزات (الانتقال الى مستوى جديد، او كسب شارات) ستجعل عملية التعليم ممتعة بالنسبة لك؟	A6: بالطبع ستجعل التعليم ممتعا وتحفيزي اكثر
Q7: هل هناك أي ميزات معينة ترغبين في رؤيتها في منصة تعليم الاستثمار؟	A7: تعليم تفاعلي مباشر، أيضا متابعة لمستوى تقدمي في المحتوى التعليمي

Semester-2 1446H (Spring 2025)

Table 12: Second Interview Outline

Interview#2	
Interview Date:	Jan 23, 2025
Start Time:	3:11 PM
End Time:	3:39 PM
Location:	The interviewer's house, Riyadh
Reminders:	<ul style="list-style-type: none"> The Interviewee's age is 24 The Interviewee is an employee
Questions:	Answers:
Q1: هل جربت الاستثمار من قبل؟	A1: لا
Q2: هل لديك دافع لتعلم الاستثمار؟ اذا كان جوابك بنعم ما هي دوافعك؟	A2: بالطبع، أرغب بتنويع مصادر الدخل، وزيادة ثقافي المالية بالاستثمار
Q3: ما هي الصعوبات التي تواجهك في تعلم الاستثمار؟	A3: بسبب تنوع وكثرة المنصات الموجودة حالياً لتعليم الاستثمار لم اعد اعلم ابن اجد المعلومة الصحيحة، أيضاً الخوف من الخسارة بسبب قلة الخبرة وعدم التجربة
Q4: كيف تقضلين ان تكون طريقة تعليم الاستثمار؟	A4: افضل ان تكون جلسات عن بعد، بحيث يتسعني لي السؤال، أيضاً افضل ان تكون مقاطع قصيرة
Q5: هل سبق لك تجربة استخدام تطبيقات محاكاة لتعلم الاستثمار؟ ان كانت اجابتك نعم، كيف كانت تجربتك؟	A5: لا، لم يسبق لي بتجربتها
Q6: هل وجود الألعاب والمحفزات (كالانتقال إلى مستوى جديد، او كسب شارات) ستجعل عملية التعليم ممتعة بالنسبة لك؟	A6: نوعاً ما، افضل أيضاً ان تكون الألعاب متنوعة لجذب الانتباه
Q7: هل هناك أي ميزات معينة ترغبين في رؤيتها في منصة تعليم الاستثمار؟	A7: وجود مرشد استطيع سؤاله

Semester-2 1446H (Spring 2025)

Table 13: Third Interview Outline

Interview#3	
Interview Date: Jan 23, 2025 Start Time: 8:22 PM End Time: 8:52 PM	Reminders: <ul style="list-style-type: none"> The Interviewee's age is +45 The Interviewee is a homemaker
Questions:	Answers:
Q1: هل جربت الاستثمار من قبل؟	A1: لا، أرحب بتجربته، لكن لم تتح لي الفرصة لأنني افتقر إلى الخبرة الكافية في هذا المجال
Q2: هل لديك دافع لتعلم الاستثمار؟ اذا كان جوابك بنعم ما هي دوافعك؟	A2: نعم، زيادة الدخل وتطوير معرفتي المالية
Q3: ما هي الصعوبات التي تواجهك في تعلم الاستثمار؟	A3: قلة الخبرة في تحديد ما إذا كان المصدر موثوق أم لا، الخوف من الاحتيال بسبب تنوع وتعدد المنصات والمرشدين في هذا المجال، أيضاً الخوف من الخسارة المالية
Q4: كيف تقضلين ان تكون طريقة تعليم الاستثمار؟	A4: افضل ان تكون مقاطع فيديو قصيرة تشرح وتركز على المفاهيم الأساسية للاستثمار
Q5: هل سبق لك تجربة استخدام تطبيقات محاكاة لتعلم الاستثمار؟ ان كانت اجابتك نعم، كيف كانت تجربتك؟	A5: لا، لم يسبق لي بتجربتها
Q6: هل وجود الألعاب والمحفزات (كالانتقال إلى مستوى جديد، او كسب شارات) ستجعل عملية التعليم ممتعة بالنسبة لك؟	A6: نعم، سوف يصبح التعليم محفز
Q7: هل هناك أي ميزات معينة ترغبين في رؤيتها في منصة تعليم الاستثمار؟	A7: افضل اختيار مدة معينة للتعليم وبناء على المدة الزمنية يكون هناك تتبع لتقديمي التعليمي

Semester-2 1446H (Spring 2025)

Table 14: Fourth Interview Outline

Interview#4	
Interview Date:	Jan 25, 2025
Start Time:	6:37 PM
End Time:	6:51 PM
Location:	Coffee shop, Riyadh
Reminders:	<ul style="list-style-type: none"> The Interviewee's age is 28 The Interviewee is an employee
Questions:	Answers:
Q1: هل جربت الاستثمار من قبل؟	A1: نعم في أكثر من مجال
Q2: هل لديك دافع لتعلم الاستثمار؟ اذا كان جوابك بنعم ما هي دوافعك؟	A2: بالطبع، لزيادة الدخل
Q3: ما هي الصعوبات التي تواجهك في تعلم الاستثمار؟	A3: تنوع المصادر التعليمية والتعليم النظري الملل، أيضاً الاخذ بوصيات الخبراء دون خوض تجربة محاكية
Q4: كيف تفضلين ان تكون طريقة تعليم الاستثمار؟	A4: فيديوهات ونصوص كلا الطريقيتين مناسبة، الفيديوهات لشرح الأفكار المعقدة والنصوص للتخصيص للأفكار
Q5: هل سبق لك تجربة استخدام تطبيقات محاكاة لتعلم الاستثمار؟ ان كانت اجابتك نعم، كيف كانت تجربتك؟	A5: لم يسبق لي التجربة
Q6: هل وجود الألعاب والمحفزات (الانتقال الى مستوى جديد، او كسب شارات) ستجعل عملية التعليم ممتعة بالنسبة لك؟	A6: نعم، ستحفز دافع التعليم لدى وستزيد الفضول والرغبة بالنجاح والانتقال الى مراحل اخرى
Q7: هل هناك أي ميزات معينة ترغبين في رؤيتها في منصة تعليم الاستثمار؟	A7: إضافة محاكاة للاستثمار وتوفير تقارير شهرية لمعرفة تقدمي بالاستثمار واصافة توصيات نقل من الخسارة

13.5 Appendix B: Questionnaire

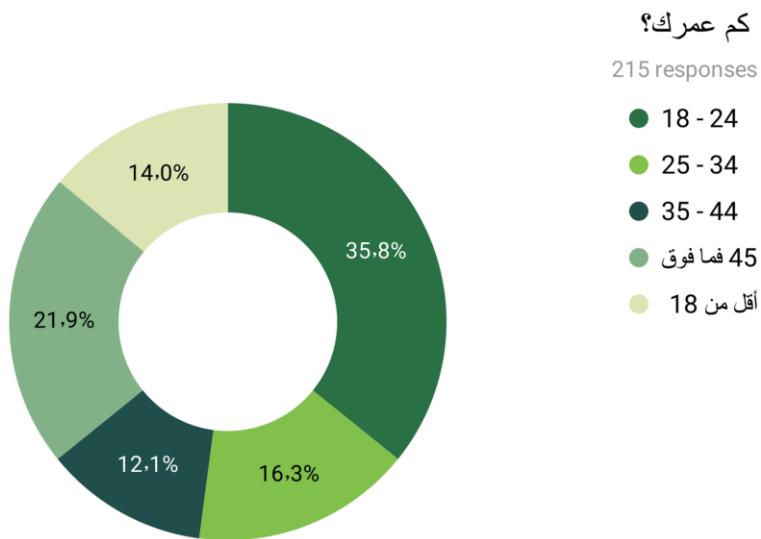


Figure 62: Pie chart of question 1 (age distribution)

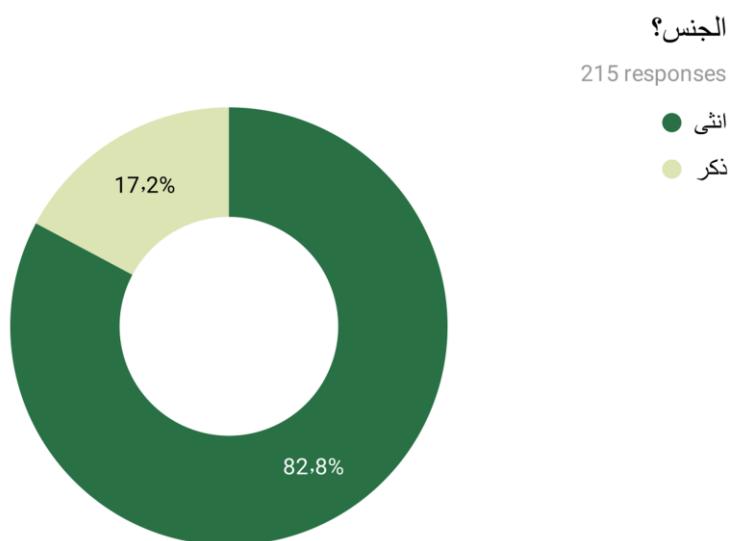


Figure 63: Pie chart of question 2 (gender distribution)

Semester-2 1446H (Spring 2025)

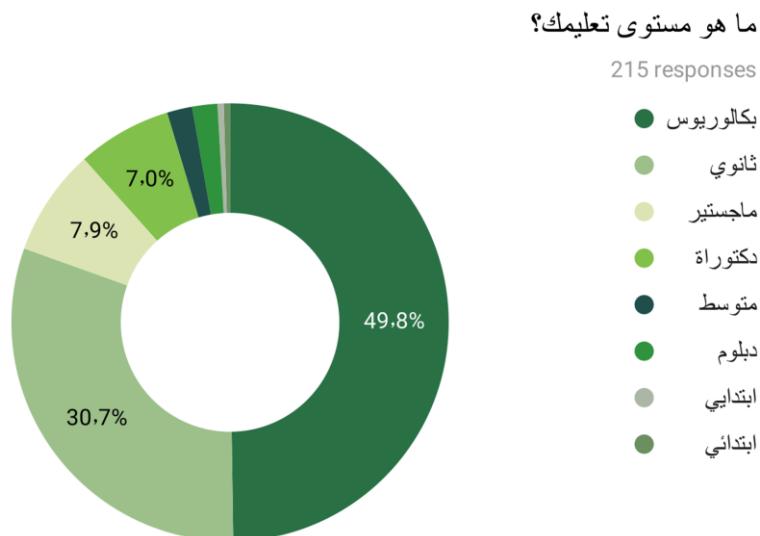


Figure 64: Pie chart of Question 3 (Education Level distribution)

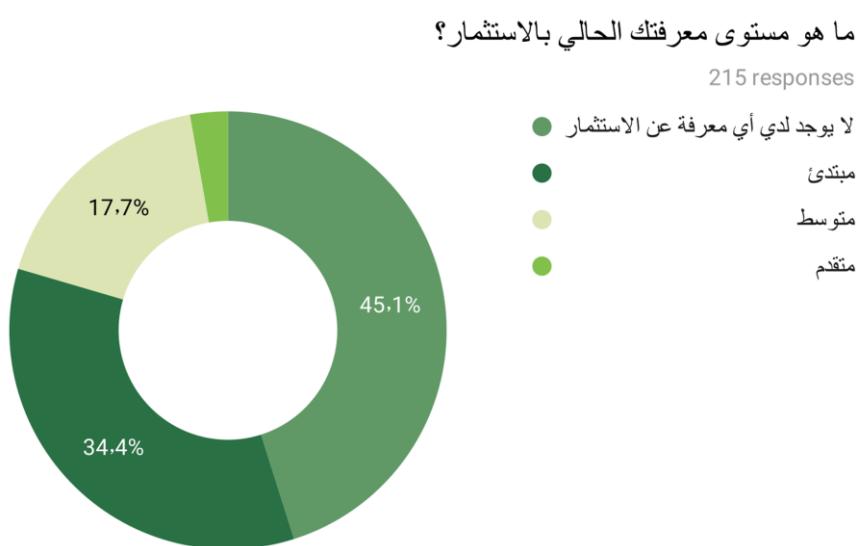


Figure 65: Pie chart of Question 4 (Current Investment Knowledge distribution)

Semester-2 1446H (Spring 2025)

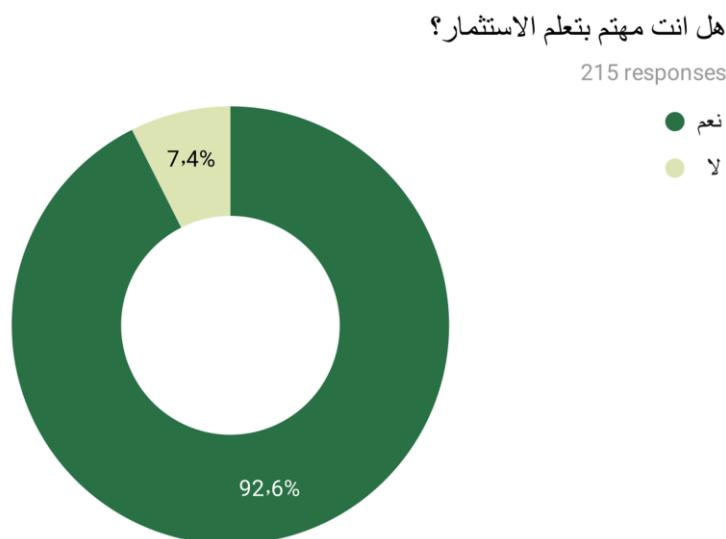


Figure 66: Pie chart of Question 5 (Interest in Learning Investment distribution)

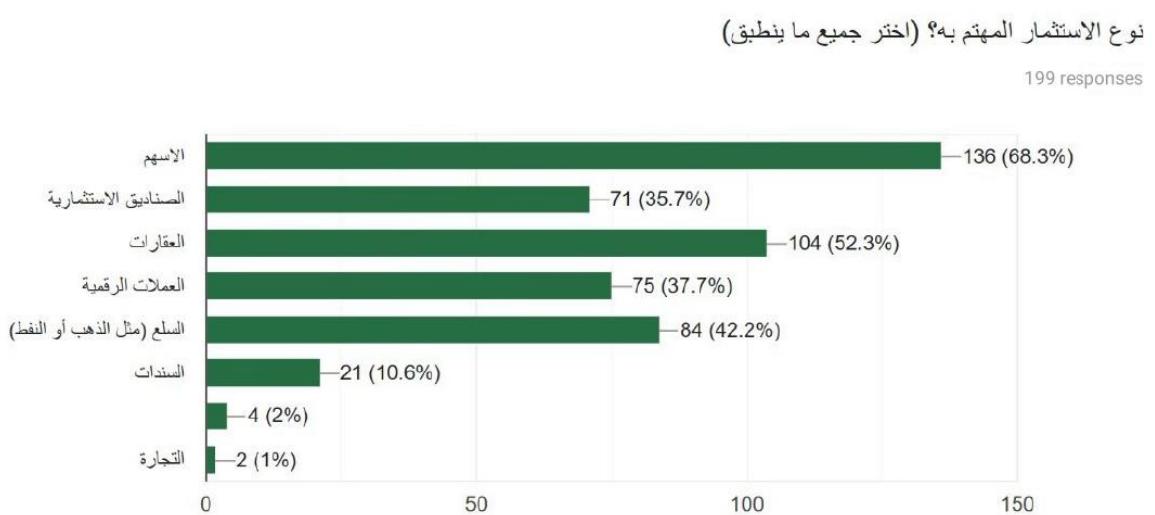


Figure 67: Bar chart of Question 6 (Preferred Investment Type distribution)

Semester-2 1446H (Spring 2025)

كلية علوم الحاسوب والمعلومات
 قسم تكنولوجيا المعلومات

ما هو دافعك لتعلم الاستثمار؟ (اختر جميع ما ينطبق)

199 responses

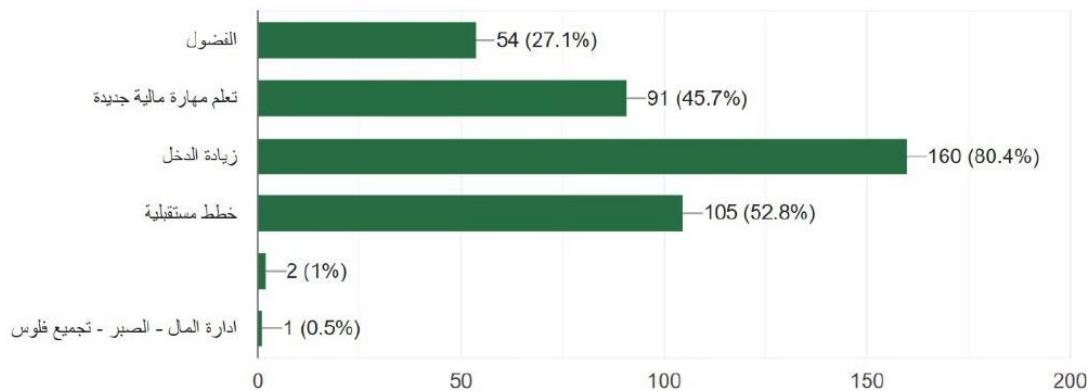


Figure 68: Bar Chart of Question 7 (Motivation for Learning Investment distribution)

ما هي التحديات التي تواجهها عند تعلم الاستثمار؟ (اختر جميع ما ينطبق)

199 responses

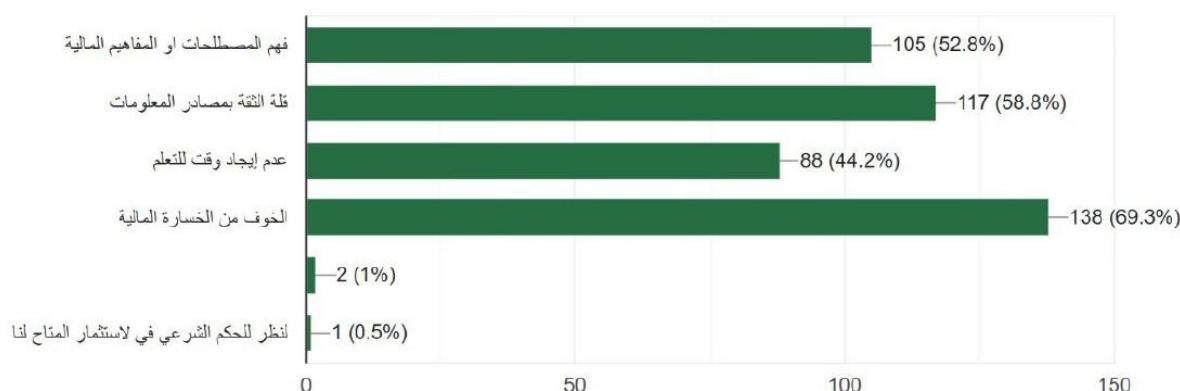


Figure 69: Bar Chart of Question 8 (Challenges Faced in Learning Investment distribution)

هل وجود منصة لتعلم ومحاكاة الاستثمار يشجعك على البدء في الاستثمار؟

215 responses

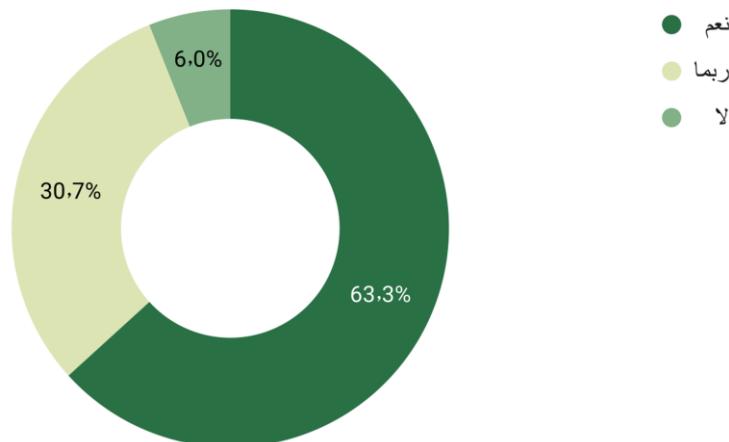


Figure 70: Pie Chart of Question 9 (Encouragement to Start Investing via Learning & Simulation)

ما هي طريقة المفضلة لتعلم الاستثمار؟

215 responses

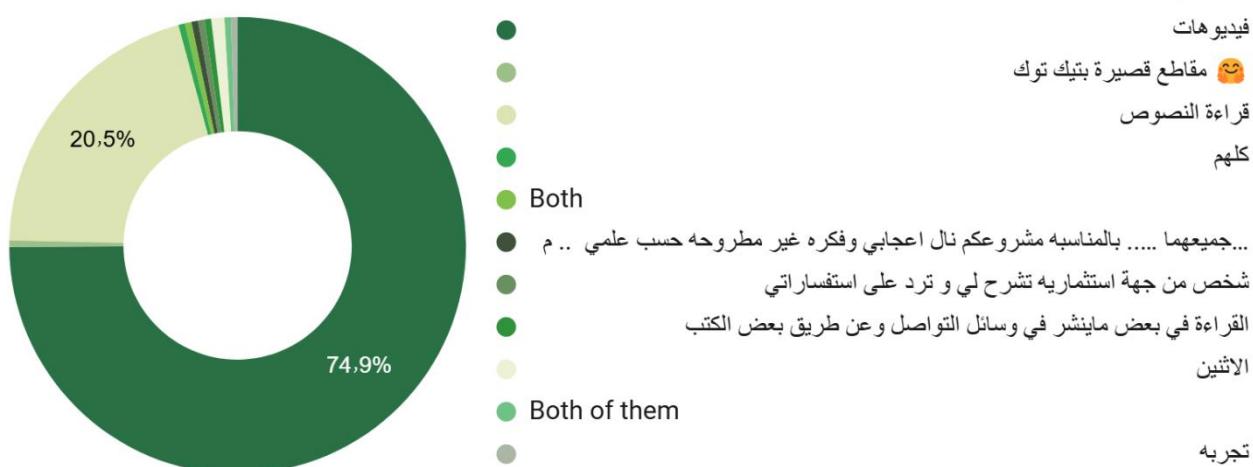


Figure 71: Pie Chart of Question 10 (Preferred Method for Learning Investment distribution)

هل تفضل استخدام موقع الكتروني او تطبيق لتعلم الاستثمار؟

215 responses

موقع الكتروني

تطبيق

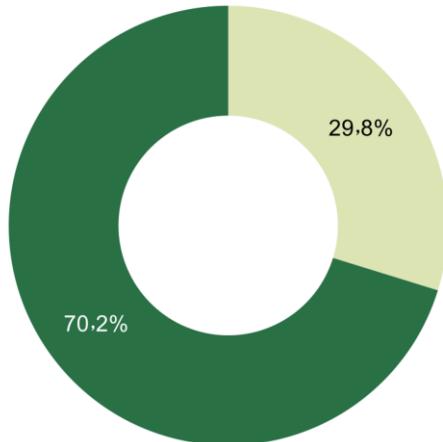


Figure 72: Pie Chart of Question 11 (Preference for Learning via Website or Application)

هل تفضل ان تكون المنصة باللغة العربية أم الانجليزية؟

215 responses

العربية

الإنجليزية

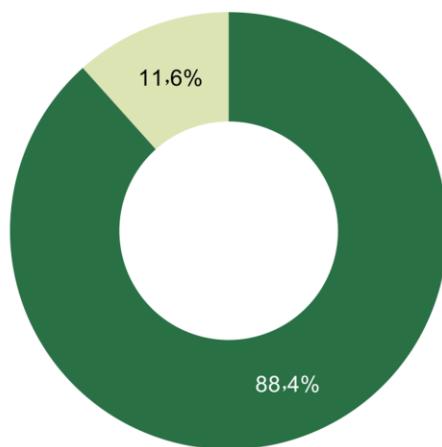


Figure 73: Pie Chart of Question 12 (Preference for Platform Language distribution)

هل يشجعك وجود تجربة شبيهة بالألعاب في المنصة على إكمال رحلتك لتعلم الاستثمار؟

215 responses

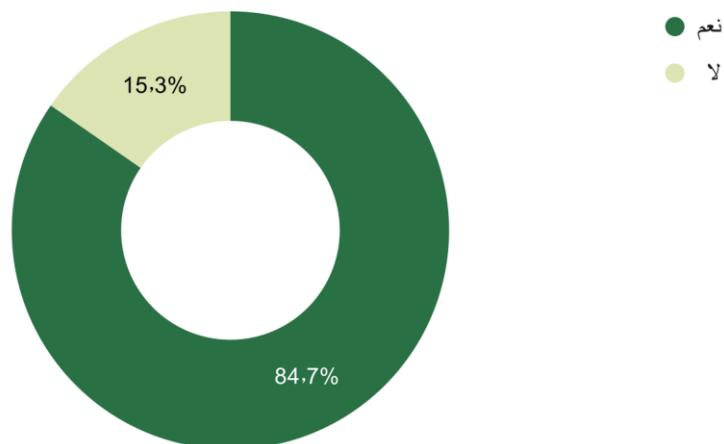


Figure 74: Pie Chart of Question 13 (Impact of Game-like Experience on Learning Continuation)

Semester-2 1446H (Spring 2025)

13.6 Appendix C: UAT Questionnaire

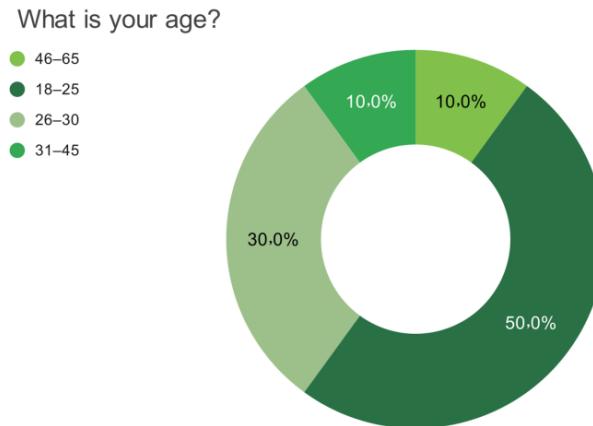


Figure 75: Age distribution

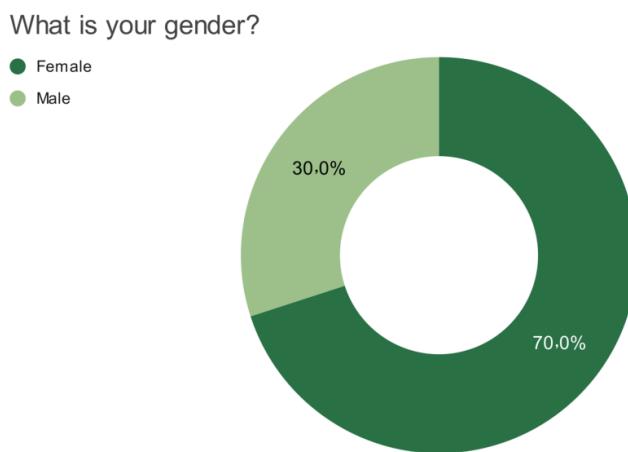


Figure 76: Gender distribution

Semester-2 1446H (Spring 2025)

What is your highest level of education completed?

- Bachelor's Degree
- High School

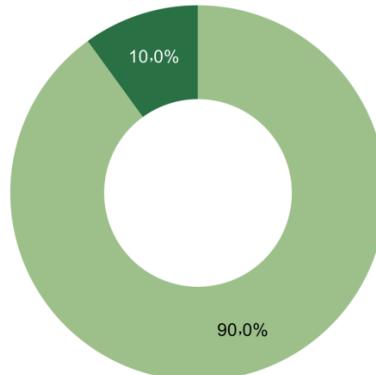


Figure 77: Education Levels

Have you used investment or finance-related apps before?

- No
- Yes

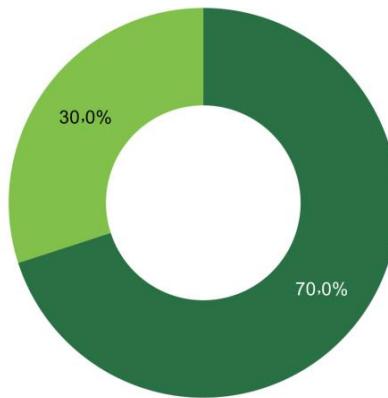


Figure 78: Prior Experience with Finance Applications

I found the Nomu app easy to navigate

- 5

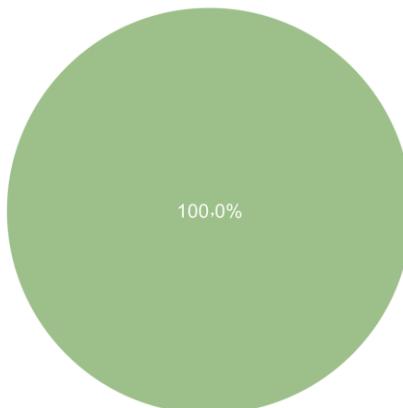


Figure 79: Ease of Navigation

Semester-2 1446H (Spring 2025)

When I made mistakes, the app showed clear and timely error messages that helped me correct them

- 3
- 4
- 5

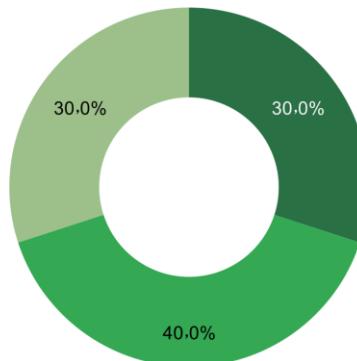


Figure 80: Error Message Clarity

The daily questions and flashcards improved my financial knowledge

- 3
- 4
- 5

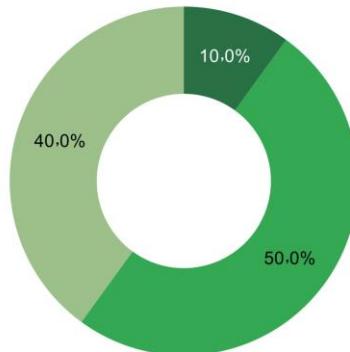


Figure 81: Flashcards & Daily Questions Impact

I was able to understand and use most features of the app without needing much time or external help

- 4
- 5

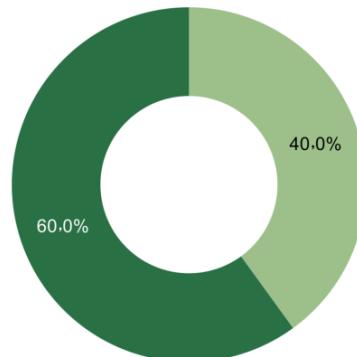


Figure 82: Feature Understanding

Semester-2 1446H (Spring 2025)

I was able to recognize icons and labels within a few seconds without confusion

- 4
- 5

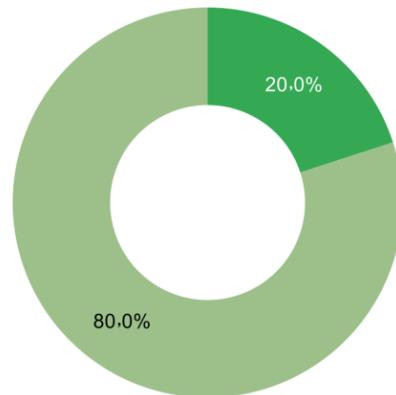


Figure 83: Icon and Label Recognition

I think I would like to use this system frequently

- 5

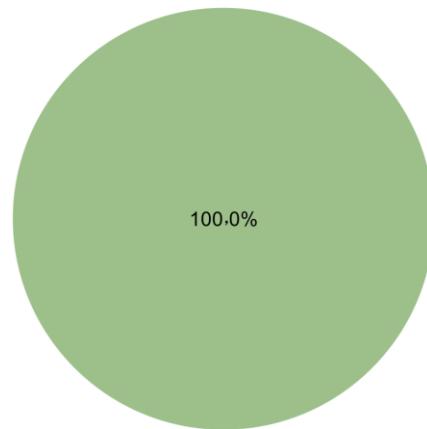


Figure 84: Willingness to Reuse the System

Semester-2 1446H (Spring 2025)

13.7 Appendix D: UAT Questionnaire

Table 15: First Interview Outline

Interview#1	
Interview Date: Apr 23, 2025 Start Time: 9:03 PM End Time: 9:17 PM	
Reminders:	
<ul style="list-style-type: none"> The Interviewee's age is 24 The Interviewee is a student. The Interviewee is a beginner in investing. 	
Questions:	Answers:
Q1: هل كان من السهل العثور على المواد التعليمية (الفيديوهات التعليمية، البطاقات التفاعلية، سؤال التحدي اليومي) وفهمها؟ ولماذا؟	A1: نعم، لقد أحببت طريقة التنظيم والمعلومات كانت واضحة.
Q2: ما رأيك في نظام التحفيز بالنقاط (الحصول على عملات)؟ هل شجعك على الاستمرار في التعلم؟	A2: نعم إنها مشجعة، أراها إضافة جيدة للتطبيق.
Q3: هل واجهت أي شيء أربك أو كان صعب الاستخدام؟	A3: لا، كل شيء واضح بالنسبة لي.
Q4: إذا كان بإمكانك تغيير أو إضافة شيء واحد لتحسين التطبيق، ماذا سيكون؟	A4: أنا أراه جيد ومتكملاً . لا أرى أنه يحتاج إلى أي شيء.
Q5: هل تشعر بالثقة عند استخدام هذا التطبيق لتعلم الاستثمار؟	A5: نعم أثق فيه، لمصادر المعلومات الموثوقة.
Q6: هل ستتصفح الآخرين باستخدام هذا التطبيق؟	A6: نعم، بالتأكيد.

Table 16: Second Interview Outline

Interview#2	
Questions:	Answers:
Q1: هل كان من السهل العثور على المواد التعليمية (الفيديوهات التعليمية، البطاقات التفاعلية، سؤال التحدي اليومي) وفهمها؟ ولماذا؟	A1: نعم، كان كل شيء واضح.
Q2: ما رأيك في نظام التحفيز بالنقاط (الحصول على عملات)؟ هل شجعك على الاستمرار في التعلم؟	A2: نعم ، إنها مشجعة على التعلم .
Q3: هل واجهت أي شيء أربكك أو كان صعب الاستخدام؟	A3: لا، كل شيء كان واضحا وسهلا لفهم.
Q4: إذا كان بإمكانك تغيير أو إضافة شيء واحد لتحسين التطبيق، ماذا سيكون؟	A4: أنا أراه جيد للغاية، لكنني ربما سأدعم لغات أخرى.
Q5: هل تشعر بالثقة عند استخدام هذا التطبيق لتعلم الاستثمار؟	A5: أنا أثق بالمعلومات المقدمة في المواد التعليمية، أما عن المحاكاة فلا استطيع ان اعطي رأيي حاليا إلا عند تجريبه.
Q6: هل ستتصفح الآخرين باستخدام هذا التطبيق؟	A6: نعم، بالتأكيد لأنه واضح وشامل.

Table 17: Third Interview Outline

Interview#3	
Questions:	Answers:
Q1: هل كان من السهل العثور على المواد التعليمية (الفيديوهات التعليمية، البطاقات التفاعلية، سؤال التحدى اليومي) وفهمها؟ ولماذا؟	A1: نعم، أحببت طريقة تنظيم المعلومات وطرق عرضها فقد كانت واضحة وموجزة ومقدمة بطرق شبيهة.
Q2: ما رأيك في نظام التحفيز بالنقاط (الحصول على عملات)؟ هل شجعك على الاستمرار في التعلم؟	A2: نعم، لقد أحببته أراها إضافة جيدة.
Q3: هل واجهت أي شيء أرباك أو كان صعب الاستخدام؟	A3: لا، أراه واضحاً وسهلاً الاستخدام.
Q4: إذا كان بإمكانك تغيير أو إضافة شيء واحد لتحسين التطبيق، ماذا سيكون؟	A4: سأقوم بدعم لغات أخرى أيضاً.
Q5: هل تشعر بالثقة عند استخدام هذا التطبيق لتعلم الاستثمار؟	A5: نعم أثق فيه.
Q6: هل ستتصحّح الآخرين باستخدام هذا التطبيق؟	A6: نعم، سأناصح الآخرين باستخدامه.

Table 18: Forth Interview Outline

Interview#4	
Questions:	Answers:
Q1: هل كان من السهل العثور على المواد التعليمية (الفيديوهات التعليمية، البطاقات التفاعلية، سؤال التحدي اليومي) وفهمها؟ ولماذا؟	A1: نعم، كان كل شيء واضح والمحتوى التعليمي مبسط وشيق.
Q2: ما رأيك في نظام التحفيز بالنقاط (الحصول على عملات)؟ هل شجعك على الاستمرار في التعلم؟	A2: نعم إنها مفيدة ومحفزة.
Q3: هل واجهت أي شيء أربكك أو كان صعب الاستخدام؟	A3: لا، حاليا كل شيء واضح.
Q4: إذا كان بإمكانك تعديل أو إضافة شيء واحد لتحسين التطبيق، ماذا سيكون؟	A4: ربما إضافة شبكة تواصل مع الأصدقاء والخبراء داخل التطبيق.
Q5: هل تشعر بالثقة عند استخدام هذا التطبيق لتعلم الاستثمار؟	A5: نعمأشعر بالثقة بالمحتوى التعليمي.
Q6: هل ستتصحّح الآخرين باستخدام هذا التطبيق؟	A6: بالطبع سأتصحّح الجميع باستخدامه.

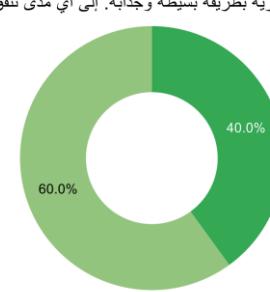
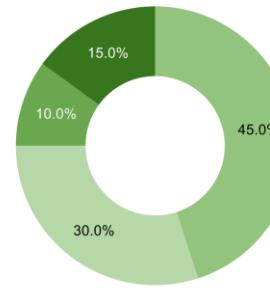
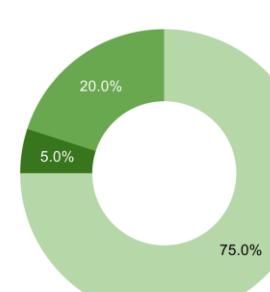
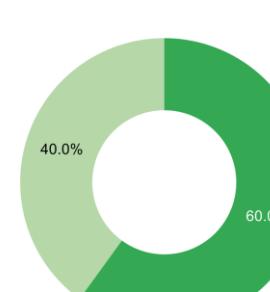
Table 19: Fifth Interview Outline

Interview#5	
Questions:	Answers:
Q1: هل كان من السهل العثور على المواد التعليمية (الفيديوهات التعليمية، البطاقات التفاعلية، سؤال التحدى اليومي) وفهمها؟ ولماذا؟	A1: نعم، كان كل شيء واضح ومقدم بطرق شيقة وبسيطة والانتقال كان سلساً.
Q2: ما رأيك في نظام التحفيز بالنقاط (الحصول على عملات)؟ هل شجعك على الاستمرار في التعلم؟	A2: أرى أنها جيدة ومشجعة على التعلم.
Q3: هل واجهت أي شيء أرباك أو كان صعب الاستخدام؟	A3: حاليا كل شيء واضح، لكن اتوقع لتجربة نظام المحاكاة وال töصيات.
Q4: إذا كان بإمكانك تغيير أو إضافة شيء واحد لتحسين التطبيق، ماذا سيكون؟	A4: دعم لغات أخرى وتوفير روبوت الدردشة للإجابة على الأسئلة.
Q5: هل تشعر بالثقة عند استخدام هذا التطبيق لتعلم الاستثمار؟	A5: على الأغلب نعم لمصادر التعلم الموثوقة.
Q6: هل ستتصفح الآخرين باستخدام هذا التطبيق؟	A6: بالطبع سأناصح الجميع باستخدامه.

Semester-2 1446H (Spring 2025)

13.8 Appendix E: Objectives evaluation Questions and Results

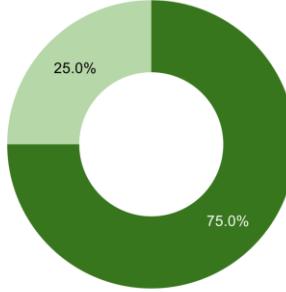
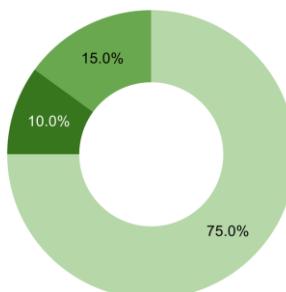
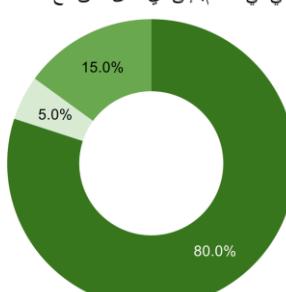
Table 20: objectives evaluation Questions and Results

objectives evaluation Questions and Results											
Question	Chart and results										
1. The learning tools in Nomu (daily challenge, educational videos, and interactive flashcards) helped me understand investment concepts in a simple and engaging way. To what extent do you agree with this statement?	<p>ساهنت أدوات التعلم في تطبيق نمو (السؤال اليومي، مقاطع الفيديو التعليمية، والبطاقات التفاعلية) في فهمي للمفاهيم الاستثمارية بطريقة بسيطة وجذابة. إلى أي مدى تتفق مع هذه العبارة؟</p>  <table border="1"> <thead> <tr> <th>Response</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td> موافق بشدة</td> <td>60.0%</td> </tr> <tr> <td> موافق</td> <td>40.0%</td> </tr> </tbody> </table>	Response	Percentage	موافق بشدة	60.0%	موافق	40.0%				
Response	Percentage										
موافق بشدة	60.0%										
موافق	40.0%										
2. The rewards system in Nomu (coins) motivated me to continue learning and completing activities. To what extent do you agree with this statement?	<p>حفزني نظام المكافآت (نقاط) في تطبيق نمو على الاستثمار في التعلم وإكمال الأنشطة.</p> <p>إلى أي مدى تتفق مع هذه العبارة؟</p>  <table border="1"> <thead> <tr> <th>Response</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td> موافق بشدة</td> <td>45.0%</td> </tr> <tr> <td> موافق</td> <td>30.0%</td> </tr> <tr> <td> غير موافق</td> <td>15.0%</td> </tr> <tr> <td> معارض</td> <td>10.0%</td> </tr> </tbody> </table>	Response	Percentage	موافق بشدة	45.0%	موافق	30.0%	غير موافق	15.0%	معارض	10.0%
Response	Percentage										
موافق بشدة	45.0%										
موافق	30.0%										
غير موافق	15.0%										
معارض	10.0%										
3. The stock market simulation in Nomu provided a realistic and easy-to-understand experience of trading. To what extent do you agree with this statement?	<p>قدمت محاكاة سوق الأسهم في تطبيق نمو تجربة واقعية وسهلة الفهم للتداول. إلى أي مدى تتفق مع هذه العبارة؟</p>  <table border="1"> <thead> <tr> <th>Response</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td> موافق بشدة</td> <td>75.0%</td> </tr> <tr> <td> موافق</td> <td>20.0%</td> </tr> <tr> <td> محيلا</td> <td>5.0%</td> </tr> </tbody> </table>	Response	Percentage	موافق بشدة	75.0%	موافق	20.0%	محيلا	5.0%		
Response	Percentage										
موافق بشدة	75.0%										
موافق	20.0%										
محيلا	5.0%										
4. The portfolio page helped me track my profits, losses, and overall performance clearly. To what extent do you agree with this statement?	<p>ساعدتني صفحة المحفظة في تتبع أرباحي وخسائره وأداني العام بشكل واضح. إلى أي مدى تتفق مع هذه العبارة؟</p>  <table border="1"> <thead> <tr> <th>Response</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td> موافق بشدة</td> <td>60.0%</td> </tr> <tr> <td> موافق</td> <td>40.0%</td> </tr> </tbody> </table>	Response	Percentage	موافق بشدة	60.0%	موافق	40.0%				
Response	Percentage										
موافق بشدة	60.0%										
موافق	40.0%										

Semester-2 1446H (Spring 2025)

<p>5. The market data and charts in Nomu enhanced my understanding of stock movements. To what extent do you agree with this statement?</p>	<p>عززت بيانات السوق والمخططات البيانية في التطبيق فهمي لحركة الأسهم. إلى أي مدى تتفق مع هذه العبارة؟</p> <table border="1"> <thead> <tr> <th>Level of Agreement</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>موافق بشدة</td> <td>45.0%</td> </tr> <tr> <td>موافق</td> <td>55.0%</td> </tr> </tbody> </table>	Level of Agreement	Percentage	موافق بشدة	45.0%	موافق	55.0%				
Level of Agreement	Percentage										
موافق بشدة	45.0%										
موافق	55.0%										
<p>6. The user interface of Nomu was simple, organized, and easy to use. To what extent do you agree with this statement?</p>	<p>كانت واجهة تطبيق نمو بسيطة ومنظمة وسهلة الاستخدام. إلى أي مدى تتفق مع هذه العبارة؟</p> <table border="1"> <thead> <tr> <th>Level of Agreement</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>موافق بشدة</td> <td>95.0%</td> </tr> <tr> <td>موافق</td> <td>5.0%</td> </tr> </tbody> </table>	Level of Agreement	Percentage	موافق بشدة	95.0%	موافق	5.0%				
Level of Agreement	Percentage										
موافق بشدة	95.0%										
موافق	5.0%										
<p>7. Navigating through the Nomu application was smooth, clear, and intuitive. To what extent do you agree with this statement?</p>	<p>كان التنقل داخل تطبيق نمو سهلاً وواضحاً وبدائيًا. إلى أي مدى تتفق مع هذه العبارة؟</p> <table border="1"> <thead> <tr> <th>Level of Agreement</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>موافق بشدة</td> <td>80.0%</td> </tr> <tr> <td>موافق</td> <td>15.0%</td> </tr> <tr> <td>محايد</td> <td>5.0%</td> </tr> <tr> <td>موافق بشدة</td> <td>0.0%</td> </tr> </tbody> </table>	Level of Agreement	Percentage	موافق بشدة	80.0%	موافق	15.0%	محايد	5.0%	موافق بشدة	0.0%
Level of Agreement	Percentage										
موافق بشدة	80.0%										
موافق	15.0%										
محايد	5.0%										
موافق بشدة	0.0%										
<p>8. The application responded quickly and performed smoothly while using different features. To what extent do you agree with this statement?</p>	<p>استجاب التطبيق بسرعة وعمل بسلامة أثناء التنقل واستخدام الميزات المختلفة. إلى أي مدى تتفق مع هذه العبارة؟</p> <table border="1"> <thead> <tr> <th>Level of Agreement</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>موافق بشدة</td> <td>50.0%</td> </tr> <tr> <td>موافق</td> <td>40.0%</td> </tr> <tr> <td>محايد</td> <td>10.0%</td> </tr> <tr> <td>موافق بشدة</td> <td>0.0%</td> </tr> </tbody> </table>	Level of Agreement	Percentage	موافق بشدة	50.0%	موافق	40.0%	محايد	10.0%	موافق بشدة	0.0%
Level of Agreement	Percentage										
موافق بشدة	50.0%										
موافق	40.0%										
محايد	10.0%										
موافق بشدة	0.0%										

Semester-2 1446H (Spring 2025)

<p>9. The favorites feature helped me save and access important lessons, flashcards, or stocks easily. To what extent do you agree with this statement?</p>	<p>ساعدتني ميزة المفضلة على حفظ الدروس أو البطاقات أو الأسهم المهمة والوصول إليها بسهولة. إلى أي مدى تتفق مع هذه العبارة؟</p>  <table border="1"> <thead> <tr> <th>Response</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>موافق بشدة</td> <td>25.0%</td> </tr> <tr> <td>موافق</td> <td>75.0%</td> </tr> </tbody> </table>	Response	Percentage	موافق بشدة	25.0%	موافق	75.0%		
Response	Percentage								
موافق بشدة	25.0%								
موافق	75.0%								
<p>10. The notifications in Nomu encouraged me to stay consistent with my learning and simulation progress. To what extent do you agree with this statement?</p>	<p>كانت الإشعارات في تطبيق نمو مفيدة وشجعني على الاستمرار في التعلم والتقدم في المحاكاة. إلى أي مدى تتفق مع هذه العبارة؟</p>  <table border="1"> <thead> <tr> <th>Response</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>موافق بشدة</td> <td>20.0%</td> </tr> <tr> <td>موافق</td> <td>45.0%</td> </tr> <tr> <td>محايد</td> <td>35.0%</td> </tr> </tbody> </table>	Response	Percentage	موافق بشدة	20.0%	موافق	45.0%	محايد	35.0%
Response	Percentage								
موافق بشدة	20.0%								
موافق	45.0%								
محايد	35.0%								
<p>11. The smart assistant in Nomu provided useful guidance and clear answers to my questions. To what extent do you agree with this statement?</p>	<p>قدم لي مساعد الدردشة الذكي في تطبيق نمو ارشادات مفيدة وإجابات واضحة على أسئلتي. إلى أي مدى تتفق مع هذه العبارة؟</p>  <table border="1"> <thead> <tr> <th>Response</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>موافق بشدة</td> <td>10.0%</td> </tr> <tr> <td>موافق</td> <td>15.0%</td> </tr> <tr> <td>محايد</td> <td>75.0%</td> </tr> </tbody> </table>	Response	Percentage	موافق بشدة	10.0%	موافق	15.0%	محايد	75.0%
Response	Percentage								
موافق بشدة	10.0%								
موافق	15.0%								
محايد	75.0%								
<p>12. The Market Prediction page in Nomu provided helpful insights that improved my understanding of potential stock trends and supported my learning experience. To what extent do you agree with this statement?</p>	<p>قدمت صفحة توقعات السوق في تطبيق نمو تحليلات مفيدة حسنت فهمي لاتجاهات الأسهم المحتملة ودعمت تجربتي في التعلم. إلى أي مدى تتفق مع هذه العبارة؟</p>  <table border="1"> <thead> <tr> <th>Response</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>موافق بشدة</td> <td>5.0%</td> </tr> <tr> <td>موافق</td> <td>15.0%</td> </tr> <tr> <td>محايد</td> <td>80.0%</td> </tr> </tbody> </table>	Response	Percentage	موافق بشدة	5.0%	موافق	15.0%	محايد	80.0%
Response	Percentage								
موافق بشدة	5.0%								
موافق	15.0%								
محايد	80.0%								

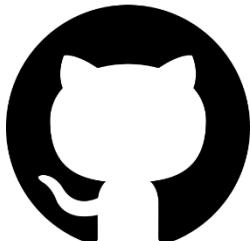
Semester-2 1446H (Spring 2025)

<p>13. Nomu increased my confidence in understanding and practicing investment concepts. To what extent do you agree with this statement?</p>	<p>زاد تطبيق نمو من تبني في ممارسة وفهم مفاهيم الاستثمار. إلى أي مدى تتفق مع هذه العبارة؟</p> <table border="1"> <thead> <tr> <th>الرأي</th> <th>نسبة (%)</th> </tr> </thead> <tbody> <tr> <td>موافق بشدة</td> <td>65.0%</td> </tr> <tr> <td>موافق</td> <td>35.0%</td> </tr> </tbody> </table>	الرأي	نسبة (%)	موافق بشدة	65.0%	موافق	35.0%
الرأي	نسبة (%)						
موافق بشدة	65.0%						
موافق	35.0%						
<p>14. Overall, I am satisfied with the Nomu application and its ability to support beginners in learning about investment. To what extent do you agree with this statement?</p>	<p>بشكل عام، أنا راضٍ عن تطبيق نمو وقدرته على دعم المبتدئين في التعلم عن الاستثمار. إلى أي مدى تتفق مع هذه العبارة؟</p> <table border="1"> <thead> <tr> <th>الرأي</th> <th>نسبة (%)</th> </tr> </thead> <tbody> <tr> <td>موافق بشدة</td> <td>10.0%</td> </tr> <tr> <td>موافق</td> <td>90.0%</td> </tr> </tbody> </table>	الرأي	نسبة (%)	موافق بشدة	10.0%	موافق	90.0%
الرأي	نسبة (%)						
موافق بشدة	10.0%						
موافق	90.0%						
<p>14. What is your overall opinion of the Nomu application?</p>	<p>تطبيق ممتاز جداً سلس ومبسط ومريح للاستخدام ويوصل المعلومات بشكل واضح وينتظم المزيد من الابداع . تطبيق مفيد جداً لطلاب المالية بشكل خاص وللمجتمع بشكل عام حيث ان اللغة سهلة ويسيرة والمفاهيم مبسطة وتساعد على التحفيز خ لمواصلة التعلم اما بالنسبة لطلاب المالية فغيره التطبيق مفيد جداً لمحاركة سوق العمل السعودي والتعلم من خلال الاقتصاد المحلي بدلاً من التعلم على المطبيقات الأمريكية حلو وواضح جميل للامانة ومتذكر ، ومتذكرة يكون مستقبلاً واعد لو تفكرون تنشرونه فكرة جديدة اول مرة اشوفها ميد عين تطبيق بسيط وسهل الاستخدام ويدعم التعلم لما فيه من ميزات مثل توقعات السوق باستخدام الكتاب الاصطناعي جميل ميد عين وفكرة متذكرة تطبيق مفيد جداً وسهل الاستخدام ويسهل مفاهيم الاستثمار بشكل كبير تطبيق ممتاز وسهل الاستخدام وفريدة من نوع من الابداع تطبيق رائع وفكرة ماتتقىها سابقاً ومفيدة جداً وتحفز الشخص على التعلم واعجبني تغير الانلوان بناءاً على الارتفاع او الانخفاض ولكن يوجد اقتراحات للتطوير متنازع ، اصابة الطبقات التعليمية خطوة كبيرة تدل على انكم تهتمون ان المستخدم يأخذ اقصى استفادة تعليمية سهل ويسهل ومانفعه ظسلة كجزء من قل مبنية بالاستثمار ارى ان التطبيق جزئي للتعلم وتجربة التداول والاستثمار تطبيق يسهل لي فهم التداول ويسهل لي الدخول لمجال الاسهم ووفر لي وقت وجهد التطبيق سهل الاستخدام ويوفر طريقة واضحة ومبشرة، سعادتي على فهم حركة الأسهم بسهولة واعجبني اسلوب العرض البسيط والمناسب مفيد والتطبيق واضح ويسهل مناسب للكل اعجبني كيف انه يعلم المبتدئين الاستثمار بصورة واضحة وبالعربي لأن التطبيقات الثانية التي جربتها ما كانت بهذا الوضوح التطبيق مفيد وله مستقبل باهر تطبيق رائع ومتنازع منحصراً جداً لاستخدامه رائع اعجبني وراح استخدمه اذا تشربته</p>						

Semester-2 1446H (Spring 2025)

15. If you have any suggestions to improve the Nomu application, please mention them.	اصدافة نسخة بالإنجليزي علشان توسعون وايضاً نسخة للآيفون ممكن لو تضييقون شركات اكتر فيه شركات معروفة ما فيتها اقرخ تطوير التطبيق بدعم لغات الخرى . دعم لغات متعددة - جعل البيانات واقعية وغير تاريخية- تقليل مدة النظار صفحة التوقعات - عرض اسم الشركة بشكل كامل في صفحة التوقعات زيادة سرعة استجابة التطبيق ممكن لو تستبدلوا مقطع الفيديو بجنسات مباشرة مع خبراء في الاستثمار ان يشمل على الشركات العالمية وليس فقط على العربية باليت لو كان فيه دارك مود بيصير احسن اصدافة تحديات بين المستخدمين وعرض قائمة المتتصدين استخدام داتا حقيقة تمثل داتا الوقت الحالي الدروس كانت قبله بس مجنيبي انها قصيرة ما ملئت وانا اقرجها
---------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

13.9 Appendix F: GitHub repository and Jira project



Link: <https://github.com/asmaib/GP-Nomu.git>



Link: <https://student-team-iibdsij2.atlassian.net/jira/software/projects/SCRUM/boards/1/backlog>