**STOCK PRICE TRACKER AND UPDATER**

**A PROJECT REPORT**

***Submitted by***

|  |  |
| --- | --- |
| **UMESH SUBRAMANIAN S** | **220701307** |

***in partial fulfillment of the course***

**OAI1903 - INTRODUCTION TO ROBOTIC PROCESS AUTOMATION**

***for the degree of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**



**RAJALAKSHMI ENGINEERING COLLEGE**

**RAJALAKSHMI NAGAR**

**THANDALAM**

**CHENNAI – 602 105**

**NOVEMBER 2024**

**RAJALAKSHMI ENGINEERING COLLEGE**

**CHENNAI - 602105**

**BONAFIDE CERTIFICATE**

Certified that this project report **“STOCK PRICE TRACKER AND UPDATER”** is the bonafide work of **“UMESH SUBRAMANIAN (220701307)”** who carried out the project work for the subject OAI1903-Introduction to Robotic Process Automation under my supervision.

|  |  |
| --- | --- |
|  | **Ms. U.Farjana, M.Tech.**  **SUPERVISOR**  Assistant Professor  Department of Computer Science and Engineering  Rajalakshmi Engineering College  Rajalakshmi Nagar  Thandalam  Chennai - 602105 |

Submitted to Project and Viva Voce Examination for the subject OAI1903-Introduction to Robotic Process Automation held on \_\_\_\_\_\_\_\_\_\_.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

Initially, we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Thiru. S.Meganathan, B.E., F.I.E.,** our Vice Chairman **Mr. M.Abhay Shankar, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) Thangam Meganathan, M.A., M.Phil., Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring to educate us in their premier institution.

Our sincere thanks to **Dr. S.N.Murugesan, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. Kumar, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. **Ms. U.Farjana, M.Tech.,** Assistant Professor**,** Department of Computer Science and Engineering for their valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinators,Professor, **Dr. N.Durai Murugan, M.E., Ph.D.,** Associate Professor, and **Mr. B.Bhuvaneswaran, M.E.,** Assistant Professor (SG), Department of Computer Science and Engineering for their useful tips during our review to build our project.

**UMESH SUBRAMANIAN S (220701307)**

**ABSTRACT**

The automation of repetitive and data-intensive tasks has become a cornerstone of efficiency in modern industries. This project focuses on the development and implementation of a robotic process automation (RPA) solution using UiPath Studio to track stock prices from a financial website. The project involves retrieving stock prices at an initial timestamp, waiting for a predefined interval, fetching the updated prices, and recording the changes in a structured Excel spreadsheet.

The objective of this project is to demonstrate the power and flexibility of RPA tools in automating real-time data acquisition and processing. The automation leverages UiPath Studio's advanced capabilities, such as web scraping, data manipulation, and seamless integration with Microsoft Excel. These features enable the system to handle dynamic web elements and process large datasets efficiently, making it a viable solution for financial analysis and stock tracking.

Key components of the project include a systematic workflow for extracting stock prices, a robust data comparison logic to identify changes, and a user-friendly output in Excel that highlights the updated values. The automation eliminates manual intervention, reduces the risk of human errors, and significantly improves data processing speed.

In this report, we outline the step-by-step methodology adopted in designing the workflow, the technical challenges encountered during implementation, and the solutions employed to overcome them. The results showcase a reliable and efficient automation process capable of handling dynamic data and producing real-time insights.

This project serves as a prototype for broader applications in financial markets, where real-time data monitoring and processing are crucial. Potential future enhancements, such as integration with stock market APIs, can further improve reliability and scalability.

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

**ABSTRACT iv**

**LIST OF FIGURES v**

**LIST OF ABBREVIATIONS vi**

**1. INTRODUCTION 1**

1.1 BACKGROUND 1

1.2 PROBLEM STATEMENT 1

1.3 PROJECT OBJECTIVES 2

1.4 SCOPE OF THE PROJECT 2

1.5 LIMITATIONS 3

**2. LITERATURE REVIEW 4**

2.1 GENERAL 4

2.2 STATE OF THE ART TECHNIQUES 4

**3. SYSTEM DESIGN 6**

3.1 SYSTEM FLOW DIAGRAM 6

3.2 ARCHITECTURE DIAGRAM 7

3.3 SOFTWARE AND HARDWARE 8

REQUIREMENTS

**4. PROJECT DESCRIPTION 9**

4.1 METHODOLOGIES 9

**5. IMPLEMENTATION AND RESULT 11**

5.1 IMPLEMENTATION PROCEDURE 11

5.2 OUTPUT 12

5.3 RESULT AND ANALYSIS 15

**6. CONCLUSION AND FUTURE WORK 20**

6.1 CONCLUSION 20

6.2 FUTURE WORKS 21

**REFERENCES 22**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **FIGURE NAME** | **PAGE NO.** |
| 3.1 | SYSTEM FLOW DESIGN | 4 |
| 3.2 | ARCHITECTURE DIAGRAM | 5 |
| 5.2 | WORKFLOW | 10 |
| 5.2 | SAMPLE INPUT 1 | 10 |
| 5.2 | SAMPLE OUTPUT 1 | 11 |
| 5.2 | SAMPLE OUTPUT 2 | 11 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATION** | **ACRONYM** |
| B.E. | Bachelor of Engineering |
| M.Tech. | Master of Technology |
| Ph.D. | Doctor of Philosophy |
| RPA | Robotic Process Automation |
| URL | Uniform Resource Locator |
| API | Application Programming Interface |
| CSV | Comma-Separated Values |
| MS Excel | Microsoft Excel |
| GUI | Graphical User Interface |
| OCR | Optical Character Recognition |

## CHAPTER 1

## INTRODUCTION

1.1 BACKGROUND

In today’s dynamic financial environment, real-time tracking of stock prices is crucial for investors, analysts, and traders who rely on this data to make informed decisions. The stock market is highly volatile, and prices can fluctuate rapidly based on various factors such as market trends, news, and global economic events. Traditionally, stock prices are monitored manually or through existing financial tools, but these methods often require continuous human intervention, are error-prone, and may lack scalability for tracking large volumes of stocks.

The emergence of Robotic Process Automation (RPA) technologies has opened up new possibilities for automating such repetitive, data-intensive tasks. RPA tools like UiPath offer a user-friendly platform for automating web scraping, data processing, and reporting. This project aims to leverage UiPath Studio to automate the process of retrieving stock prices from a financial website, comparing them over a set interval, and recording any changes in a structured format like Excel. By automating this process, the project enhances efficiency, reduces errors, and ensures accurate real-time tracking of stock prices.

**1.2 Problem Statement**

1.2 PROBLEM STATEMENT

Manually tracking stock prices is time-consuming, error-prone, and inefficient, especially when monitoring multiple stocks over time. Financial professionals need real-time data to make informed decisions, but manual methods often lead to delays and inaccuracies. An automated system is needed to efficiently track, compare, and record stock prices without human intervention. This project aims to solve this problem by automating the process of retrieving, comparing, and updating stock prices using UiPath Studio.

1.3 PROJECT OBJECTIVES

The main objectives of this project are as follows:

1. **Automate Stock Price Retrieval**: Develop an RPA solution using UiPath Studio that extracts stock prices from a financial website at predefined intervals.
2. **Compare Stock Prices**: Implement a mechanism to compare the initially fetched stock prices with subsequent prices after a set time period.
3. **Record Data in Excel**: Ensure that the updated prices and the changes in stock prices are recorded in an Excel spreadsheet, providing a clear, structured format for analysis.
4. **Enhance Efficiency and Accuracy**: Reduce human intervention in the data collection and comparison process, minimizing errors and improving overall operational efficiency.
5. **Scalability for Future Enhancements**: Create a flexible system that can be easily extended to accommodate additional stock symbols or integrated with more complex financial systems in the future.

1.4 SCOPE OF THE PROJECT

The project focuses on the automation of the following key tasks:

1. **Data Retrieval**: Extracting stock prices from a selected financial website using UiPath’s web scraping capabilities.
2. **Data Comparison**: Retrieving updated stock prices at a later point and comparing them with the initial values.
3. **Excel Integration**: Storing the extracted stock prices, including changes, in an Excel spreadsheet for future analysis and reporting.
4. **Performance Evaluation**: Ensuring that the automated process runs efficiently, with minimal manual intervention.

The project scope is limited to tracking stock prices from a single website (e.g., Yahoo Finance or Google Finance) and comparing the prices at two points in time. It does not include real-time continuous monitoring or integration with third-party APIs for direct stock data retrieval.

1.5 LIMITATIONS

While the project aims to automate stock price tracking efficiently, there are several limitations to be considered:

1. **Limited to a Single Website**: The project focuses on scraping stock data from a single website, limiting the flexibility in terms of data sources. For broader applicability, future work could involve scraping multiple websites or integrating with official stock market APIs.
2. **Manual Setup**: The system relies on the pre-configured list of stocks to track, which limits its scalability in real-time stock analysis or broad market coverage without further modification.
3. **Dynamic Web Elements**: Web scraping can be impacted by changes in the website’s structure, requiring frequent adjustments to selectors or scraping logic.
4. **Time-Dependent Data**: The current version tracks stock prices at only two points in time. Real-time or more frequent updates are not within the scope, and would require significant adjustments to the process.
5. **Excel Storage Limitations**: While the Excel output provides a convenient way to store and display data, it may not be the most efficient for handling large datasets.

**CHAPTER 2**

## LITERATURE REVIEW

### 2.1 GENERAL

### Robotic Process Automation (RPA) tools like UiPath have become essential for automating repetitive tasks in various industries, including finance. RPA allows for automation of tasks such as web scraping, data processing, and reporting, significantly reducing the need for manual effort and minimizing errors. In the finance sector, real-time stock price monitoring is a common task that benefits from automation. By using RPA to retrieve stock prices from financial websites like Yahoo Finance, users can automate data collection, compare stock prices over time, and store the results in an organized format like Excel. This improves efficiency and accuracy, helping financial professionals make timely, data-driven decisions.

### 2.2 STATE OF THE ART TECHNIQUES

## The field of RPA has evolved significantly over the past few years, with various techniques and technologies emerging to handle increasingly complex tasks. Some of the prominent techniques in the context of stock price tracking and automation include:

## Web Scraping: Web scraping is the most widely used technique for automating the retrieval of data from websites. It involves using algorithms to navigate and extract relevant data from web pages. In financial applications, web scraping is commonly used to fetch stock prices, market news, and other related data. While traditional scraping techniques rely on simple HTTP requests and HTML parsing, modern RPA tools like UiPath utilize more advanced methods, such as OCR (Optical Character Recognition) and dynamic selector handling, to deal with complex, dynamically generated web content.

## APIs for Stock Price Data: Another popular method for retrieving stock data is through APIs. Stock market APIs such as Alpha Vantage, Yahoo Finance API, and IEX Cloud provide structured, real-time stock data, which can be integrated into automated systems. These APIs offer more reliable and faster access to stock price data compared to scraping, as they bypass the need to interact with a webpage. However, APIs often come with usage limits and may require subscriptions for access to premium data.

## Machine Learning for Predictive Analysis: Machine learning techniques are increasingly being used to predict stock price trends based on historical data. Techniques like regression analysis, time series forecasting (ARIMA), and neural networks are being employed to not only track but also predict future stock prices. These techniques provide more advanced analytics beyond simple price tracking by identifying patterns and forecasting future market movements.

## Cloud-Based RPA Solutions: Cloud-based RPA platforms such as UiPath Orchestrator offer the ability to manage bots and automation workflows remotely. This enables the creation of scalable, enterprise-level automation solutions that can be used to track stock prices across multiple websites, handle large volumes of data, and integrate with various systems for reporting and analysis. Cloud RPA solutions can be particularly beneficial for financial institutions with large-scale stock monitoring needs.

## Real-Time Data Processing: While traditional automation methods involve data collection at fixed intervals, the latest advancements in RPA and cloud computing allow for real-time data processing. With real-time data processing, stock prices can be continuously monitored and analyzed without significant delays. This capability is critical for high-frequency trading and other financial operations that depend on immediate access to the latest market information.

## Blockchain for Financial Data Integrity: With the rise of blockchain technology, there has been growing interest in utilizing blockchain for ensuring data integrity in financial transactions and stock market tracking. Blockchain can be used to store financial data in a tamper-proof manner, enhancing the transparency and reliability of automated stock price tracking systems. However, this technique is still in its nascent stages and is not yet widely implemented for stock price tracking.

## These state-of-the-art techniques are shaping the future of automated stock price tracking. While web scraping remains the most accessible and widely used method, combining it with APIs, machine learning, and real-time data processing offers a more robust and scalable solution for financial automation.

## 

## 

## CHAPTER 3

## SYSTEM DESIGN

### 3.1 SYSTEM FLOW DIAGRAM

### 

FIG 3.1 SYSTEM FLOW DIAGRAM

### 

### 

### 

### 

### 

### 

### 3.2 ARCHITECTURE DIAGRAM

### 

FIG 3.2 ARCHITECTURAL DIAGRAM

**Overview**

This project aims to automate the process of tracking stock prices using UiPath Studio. The main steps involved in the project are:

1. **Data Extraction**: Using UiPath’s web scraping capabilities to retrieve stock prices from a financial website at a specific time.
2. **Data Comparison**: After a set interval, extracting the updated stock prices and comparing them with the initial values to identify any changes.
3. **Excel Integration**: Recording the initial and updated prices, along with the changes, in an Excel spreadsheet for easy analysis and reporting.

By automating these tasks, the project improves efficiency, accuracy, and reduces the manual effort required in tracking stock prices over time.

### 3.3 SOFTWARE AND HARDWARE REQUIREMENTS

**Software**

* **UiPath Studio:** To design and automate the workflow.
* **Dependencies/Packages**: UiPath packages like "UiPath.Excel.Activities", "UiPath.WebAPI.Activities", and "UiPath.System.Activities" will be used for Excel integration, web scraping, and overall automation.
* **Web Browser**: A web browser (such as Google Chrome, Firefox, or Microsoft Edge) to enable the extraction of stock price data from a financial website.
* **Microsoft Excel**: Excel will be used to store the initial and updated stock prices, along with any changes, in a structured format. This project requires Excel 2016 or later for compatibility with UiPath Excel activities.

**Hardware**

* **Standard Computer:** A standard computer with sufficient processing power and memory is sufficient.

## 

## 

## 

## 

## CHAPTER 4

## PROJECT DESCRIPTION

## 4.1 METHODOLOGIES

## The methodology followed for this project can be broken down into several stages, each essential for the successful development of the automation solution.

## 1. Requirement Analysis

## The first step involved understanding the specific needs of the project:

## Data Source: The stock prices would be extracted from a financial website (e.g., Yahoo Finance).

## Automation Scope: The task was to automate stock price retrieval at two points in time, compare the data, and update the information in Excel.

## Tools Selection: UiPath Studio was selected for its capabilities in web scraping, Excel automation, and error handling, making it an ideal tool for this type of task.

## 2. Data Collection & Web Scraping

## The core of the project lies in efficiently extracting stock prices from the chosen website. This was achieved by using UiPath's Data Scraping activity. The methodology for web scraping involved:

## Identifying the Elements: Locating the stock price data using UiPath’s built-in scraping wizards.

## Handling Dynamic Data: Financial websites often contain dynamic content, so dynamic selectors and strategies (such as using wildcards in UiPath) were applied to ensure that data extraction was reliable even if page elements changed.

## 3. Data Processing & Comparison

## Once the stock prices were extracted at the initial time point, the process involved waiting for a set interval, then scraping the updated prices. The methodology for comparison followed:

## Storing Data: The initial and updated stock prices were stored in variables and/or DataTables.

## Comparison Logic: Using If-Else conditions, the stock prices were compared to detect any price changes between the two intervals.

## Storing Results: Any differences identified were stored in variables for output purposes.

## 4. Excel Integration

## After extracting and comparing the data, the next step was to record the results in Excel:

## Excel Automation: Using UiPath’s Excel Application Scope activity, the data was written into an Excel spreadsheet.

## Formatting: The updated prices and any changes were added to the spreadsheet with proper formatting for clarity. Additionally, conditional formatting was applied to highlight significant price changes.

## 5. Error Handling & Logging

## Automation systems can face interruptions such as website access issues, data extraction failures, or timeouts. The methodology included:

## Error Handling: UiPath’s Try-Catch activities were used to catch potential errors (such as missing data or failed extractions) and handle them appropriately.

## Logging: Detailed logs were generated for every execution of the automation, helping monitor progress, track errors, and maintain transparency.

## 6. Execution and Testing

## Once the automation workflow was developed:

## Execution: The workflow was executed multiple times to test for consistency, accuracy, and reliability in data extraction and updates.

## Testing: Testing was done on different stocks to ensure that the system worked across a variety of data types, from fluctuating prices to missing values.

## 7. Final Deployment and Automation Scheduling

## Scheduling: Using UiPath Orchestrator, the process was scheduled to run at desired intervals (e.g., every few hours) to ensure that stock prices were updated consistently.

## Final Deployment: The system was deployed, and users could access the updated Excel reports after each scheduled run.

## CHAPTER 5

## IMPLEMENTATION AND RESULTS

### 5.1 IMPLEMENTATION PROCEDURE (Using UiPath Studio)

1. **Environment Setup**:
   * **Installation of UiPath Studio**: First, UiPath Studio was installed on the development machine. The appropriate version of UiPath Studio (2023 or higher) was selected to ensure compatibility with the latest activities and features.
   * **Excel Setup**: Microsoft Excel 2016 or later was installed and configured to store the stock price data. The project was designed to work with Excel worksheets for seamless data recording.
2. **Web Scraping Configuration**:
   * **Data Scraping**: The UiPath **Data Scraping** wizard was used to extract stock price information from the selected financial website (such as Yahoo Finance). Using this tool, stock prices were captured by identifying the relevant web elements such as stock symbols and their respective values.
   * **Dynamic Selectors**: Since the stock data on the website is dynamic, wildcards and reliable selectors were used to ensure that the correct elements were always extracted, regardless of changes in the web page structure.
3. **Data Storage and Comparison**:
   * **Data Table Setup**: The scraped data was initially stored in a **DataTable** to facilitate comparison between the initial and updated stock prices.
   * **Price Comparison**: Once the stock prices were retrieved at two different time intervals, they were compared using simple **If-Else** conditions to check for price changes. If the price had changed, the new price was noted, and the change was calculated.
   * **Excel Integration**: The results (initial price, updated price, and price change) were written into an Excel spreadsheet using the **Excel Application Scope** activity. Each comparison was added as a new row in the sheet.
4. **Automation Scheduling**:
   * **Running the Workflow**: After ensuring the automation worked as expected, the process was scheduled to run at specified intervals using **UiPath Orchestrator** or manually within UiPath Studio to simulate real-time price tracking.
   * **Monitoring and Logs**: Logs were generated during each run to track the automation’s performance, ensuring successful data extraction, comparison, and recording.
5. **Final Testing**:
   * Extensive testing was performed to ensure that the automation was robust, reliable, and error-free. Test cases included checking the process with different stock symbols, handling missing data, and verifying the accuracy of price updates in Excel.

## 

## 

## 5.2 OUTPUT

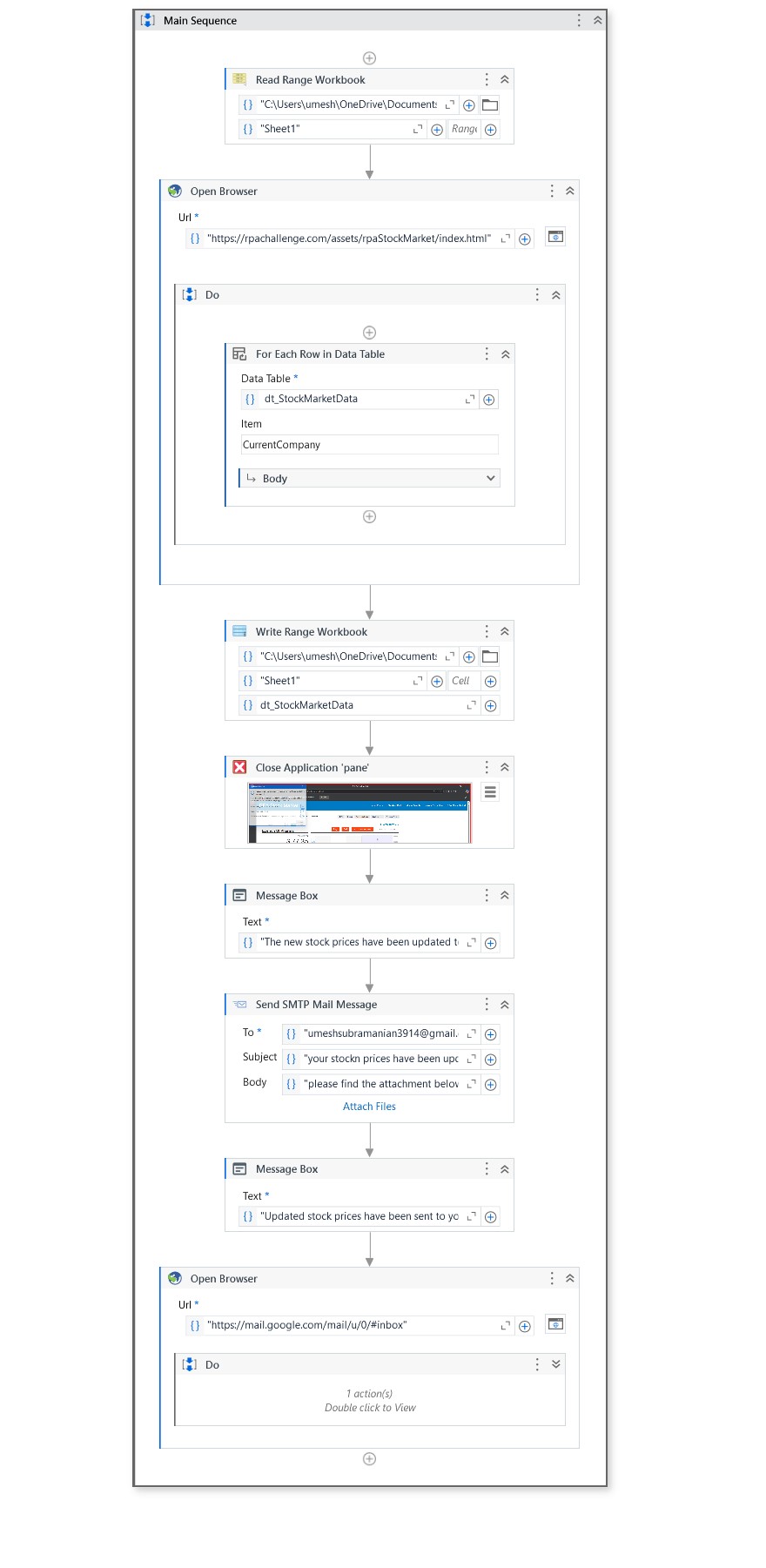


FIG 5.2 WORKFLOW

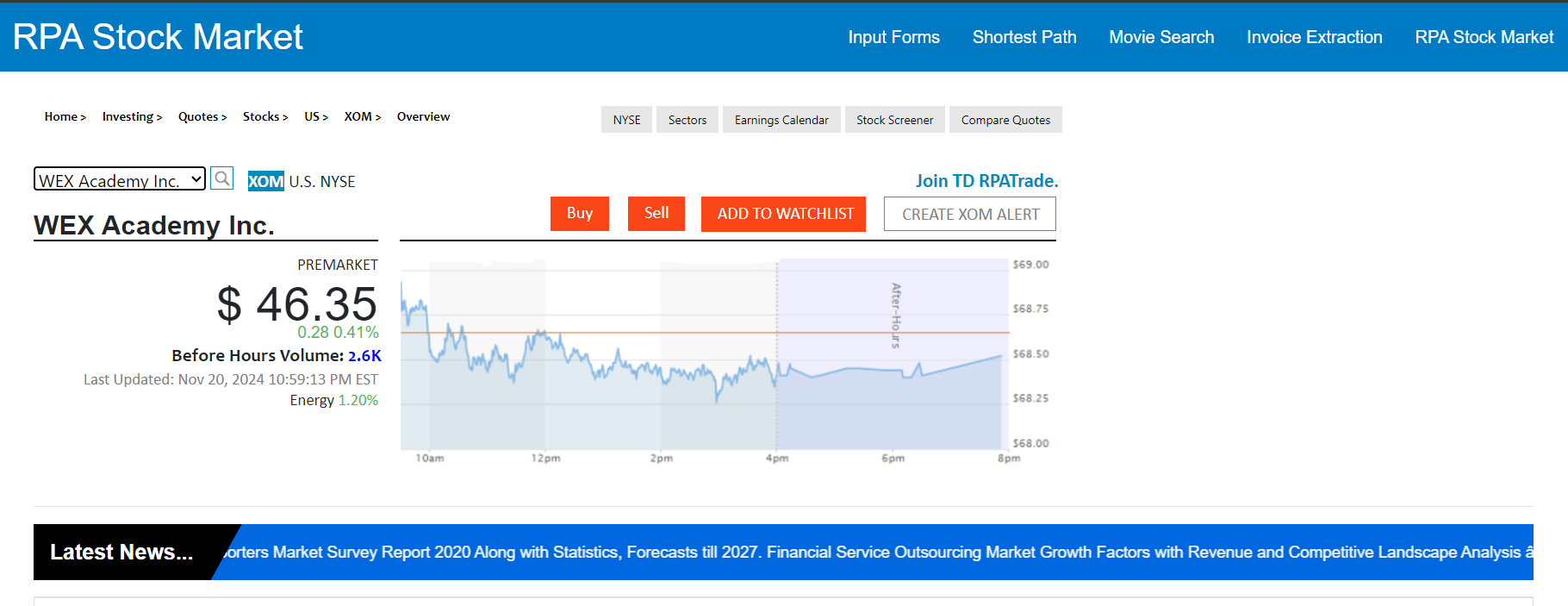


FIG 5.3 SAMPLE INPUT

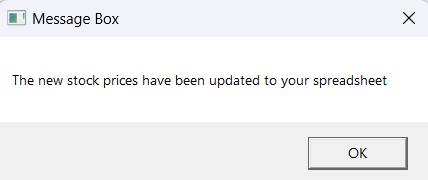
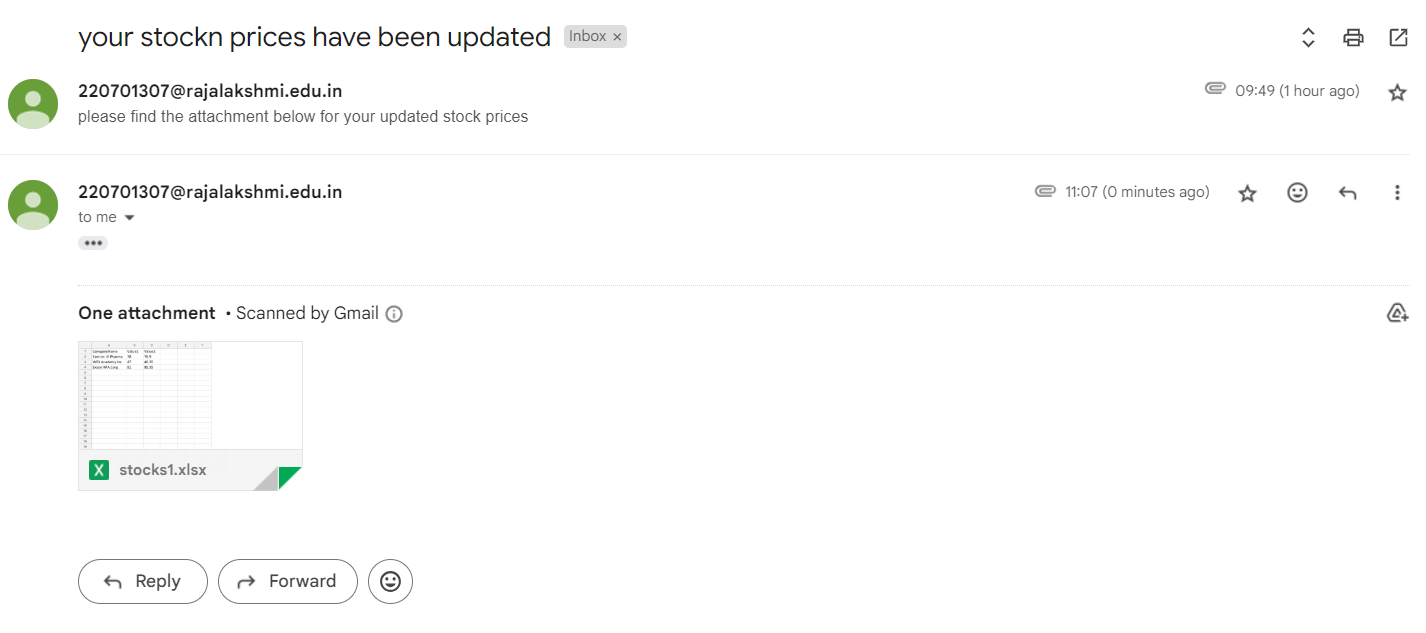


FIG 5.4 SAMPLE OUTPUT 1



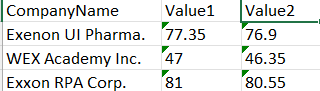


FIG 5.5 SAMPLE OUTPUT 2

## 

## 

## 5.3 RESULTS AND DISCUSSIONS

The implementation of the stock price tracking project using UiPath Studio yielded positive outcomes, achieving the desired objectives efficiently and effectively. Below is a summary of the key results:

1. **Accurate Data Retrieval**:
   * The system successfully extracted stock prices from the chosen financial website at predefined intervals. The dynamic selectors ensured that the correct data was retrieved even when minor changes occurred on the web page.
2. **Seamless Data Comparison**:
   * The system accurately compared the initial and updated stock prices, identifying price changes (increases, decreases, or no change) and recording the results.
3. **Structured Data Storage**:
   * All retrieved and computed data, including stock symbols, initial prices, updated prices, price changes, and timestamps, were stored in an Excel spreadsheet. The output was structured and clear, making it easy to analyze.
4. **Error Handling**:
   * Robust error handling mechanisms ensured that the process continued even when minor issues, such as website access delays or missing data, occurred. Logs were generated to capture details of any encountered errors.
5. **Automation Scheduling**:
   * The workflow was successfully scheduled to run automatically at specified intervals, ensuring timely updates of stock prices without requiring manual intervention.
6. **Scalability and Flexibility**:
   * The system was tested with multiple stocks and proved to be scalable for larger datasets. It can be extended to include more complex operations or integrate with additional data sources, such as APIs.

**6.2 Discussions**

The success of this project demonstrates the capabilities of Robotic Process Automation (RPA) tools like UiPath in streamlining repetitive tasks in financial environments. Several points of discussion arise from the implementation and results:

1. **Effectiveness of RPA in Financial Applications**:
   * The use of UiPath Studio to automate stock price tracking highlights the potential of RPA in the finance industry. Tasks that traditionally required significant manual effort, such as monitoring and recording stock prices, can now be automated with minimal human involvement.
2. **Advantages of Automation**:
   * The project eliminated errors typically associated with manual data entry and retrieval. By automating the process, the system ensured consistent performance, reduced time requirements, and improved the accuracy of the results.
3. **Scalability Challenges**:
   * While the project successfully handled a moderate number of stocks, managing a significantly larger dataset might introduce performance bottlenecks. To address this, optimization techniques such as parallel processing or API integration can be considered.
4. **Dependency on Web Scraping**:
   * The reliance on web scraping poses a potential limitation, as changes in the website's structure could disrupt the automation. This issue can be mitigated by using APIs provided by financial platforms, offering a more stable data retrieval mechanism.
5. **User Experience and Data Visualization**:
   * The current implementation outputs data in an Excel spreadsheet, which is suitable for small to medium-sized datasets. However, for larger-scale operations or enhanced user experience, the system could integrate with data visualization tools such as Power BI or Tableau to present the information graphically.
6. **Potential for Real-Time Monitoring**:
   * The project currently tracks stock prices at two points in time. Extending the system for real-time monitoring could add significant value for traders and analysts. This enhancement would require integrating streaming data sources or scheduling the automation to run continuously.
7. **Reliability and Maintenance**:
   * The implemented error handling mechanisms ensured reliability during testing. However, regular maintenance is necessary to adapt to changes in the financial website’s structure or updates in the UiPath Studio platform.

## 

## CHAPTER 6

## CONCLUSION AND FUTURE WORK

### 6.1 CONCLUSION

## The stock price tracking project successfully demonstrated the potential of Robotic Process Automation (RPA) in automating repetitive tasks such as retrieving, comparing, and recording stock prices. By leveraging UiPath Studio, the project achieved the following key objectives:

## Efficiently extracted real-time stock prices from a financial website.

## Accurately compared initial and updated prices to identify changes.

## Recorded and structured the results in an Excel spreadsheet for easy analysis.

## Minimized human effort and improved the accuracy of data collection and processing.

## Incorporated error handling mechanisms to ensure reliability and robustness.

## This automation significantly reduces the time and effort required for manual stock monitoring while providing accurate and consistent results. The system is scalable, user-friendly, and adaptable for use in various financial scenarios.

## 6.2 FUTURE WORK

While the current system demonstrates strong performance, there are several areas for future improvement:

While the project met its objectives, there is scope for further improvement and expansion. Below are the potential areas for future work:

1. **API Integration**:
   * Replace web scraping with API integration to improve reliability and eliminate dependency on website structures. APIs from financial platforms such as Yahoo Finance or Alpha Vantage can provide stable and direct access to stock price data.
2. **Real-Time Monitoring**:
   * Extend the automation to enable real-time stock price tracking by integrating live data feeds. This enhancement would be valuable for traders and analysts requiring up-to-the-second information.
3. **Data Visualization**:
   * Incorporate tools like Power BI or Tableau to present the tracked data in a graphical format, making it easier to analyze trends and patterns.
4. **Mobile Notifications**:
   * Add a feature to send alerts or notifications about significant price changes directly to users' mobile devices or email. This would provide real-time updates even when users are away from their systems.
5. **Scalability Enhancements**:
   * Optimize the automation to handle large datasets involving hundreds or thousands of stocks without performance bottlenecks. This can include techniques such as parallel processing and batch data handling.
6. **Historical Data Tracking**:
   * Extend the system to maintain a historical database of stock prices. This would enable trend analysis and support more informed decision-making.
7. **Integration with Trading Platforms**:
   * Connect the system with trading platforms to enable automatic trading based on predefined conditions. This would transform the system into a semi-automated trading assistant.
8. **Multi-Website Compatibility**:
   * Expand the automation to scrape data from multiple financial websites, ensuring redundancy and allowing for cross-verification of stock prices.
9. **Enhanced User Interface**:
   * Develop a custom user interface to allow users to configure stocks, intervals, and data storage preferences more intuitively.
10. **Cloud Deployment**:
    * Deploy the automation to a cloud environment, making it accessible from anywhere and enabling integration with cloud-based analytics tools.

By addressing these areas, the system can evolve into a comprehensive tool for financial monitoring and decision-making, providing even greater value to its users. The success of this project lays a strong foundation for future developments and demonstrates the transformative potential of automation in the financial domain.

## REFERENCES

1. **UiPath Documentation:  
   UiPath Studio Guide – Comprehensive documentation for UiPath Studio features and activities.**[**https://docs.uipath.com/studio**](https://docs.uipath.com/studio)
2. **Excel Integration with UiPath:  
   UiPath Official Guide on Excel Automation – Explains Excel-related activities in UiPath.**[**https://docs.uipath.com/activities/docs/excel**](https://docs.uipath.com/activities/docs/excel)
3. **Financial Data Scraping:  
   "Scraping Data from Dynamic Websites Using RPA Tools" – A practical guide to data extraction techniques using RPA.  
   Available from:** [**https://www.rpaconsultant.com**](https://www.rpaconsultant.com)
4. **Error Handling in UiPath:  
   Best Practices for Exception Handling in RPA Workflows.  
   https://www.uipath.com/blog/exception-handling-in-uipath**
5. **Yahoo Finance:  
   Financial Data Source – Stock prices and financial information.**[**https://finance.yahoo.com**](https://finance.yahoo.com)
6. **Microsoft Excel Help:  
   Official Microsoft Support for Excel – Guide to advanced Excel features used in automation.**<https://support.microsoft.com/excel>
7. **RPA Challenge:  
   Sample stock market simulation website used for this example.  
   https://rpachallenge.com/assets/rpaStockMarket/index.html**