CRYPTOCURRENCY PRICE MONITORING BOT

A PROJECT REPORT

Submitted by

PRINCE PERINBARAJ S (220701205)

in partial fulfilment for 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**"CRYPTO CURRENCY PRICE MONITORING BOT"** is the bonafide work of **"PRINCE PERINBARAJ S (220701205)"** who carried out the project work for the subject OAI1903-Introduction to Robotic Process Automation under my supervision.

SIGNATURE

Dr.N.Duraimurugan, M.E., Ph.D,
SUPERVISOR,
Assistant Professor (SG),
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**

ABSTRACT

The rapid fluctuations in cryptocurrency markets demand real-time monitoring and swift decision-making to capitalize on opportunities and minimize risks. This project introduces an Automated Cryptocurrency Price Monitoring & Alert System leveraging the Gemini API and UiPath. The system continuously tracks the prices of selected cryptocurrencies, enabling users to set customizable thresholds for alerts. When price variations surpass predefined limits, notifications are triggered in real time via email or messaging platforms, ensuring prompt user awareness.

Built using UiPath's robust automation capabilities, the system automates data retrieval, processing, and alert dissemination, offering a seamless, userfriendly interface for cryptocurrency traders and enthusiasts. By eliminating manual monitoring, this solution reduces the cognitive load on users and enhances their ability to respond proactively to market changes. The integration of the Gemini API ensures accurate and up-to-date market data, while UiPath's workflows provide scalability and ease of customization.

This project demonstrates the potential of combining RPA (Robotic Process Automation) with financial data to create intelligent, automated systems that streamline decision-making in fast-paced environments.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour 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 endeavouring in educating 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. We convey our sincere and deepest gratitude to our internal guide, **Dr.N.Durai Murugan, M.E., Ph.D.,** Associate Professor, Department of Computer Science and Engineering, Rajalakshmi Engineering College for their valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Mr.B.Bhuvaneswaran, M.E.,** Assistant Professor (SG), and Supervisor **Mrs. G.M. Sasikala, M.E., Ph.D** Department of Computer Science and Engineering for his useful tips during our review to build our project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	DEFINITION
API	Application Programming Interface
CRM	Customer Relationship Management
ERP	Enterprise Resource Planning
OCR	Optical Character Recognition
IDE	Integrated Development Environment
UML	Unified Modeling Language
UI	User Interface
LMS	Learning Management System

CHAPTER 1

INTRODUCTION

1.1 General

The rapid advancement of technology has revolutionized financial markets, particularly in cryptocurrency trading. One common challenge traders face is the constant need to monitor fluctuating cryptocurrency prices to make timely and informed decisions. This process is not only time-consuming but also mentally taxing, often resulting in missed opportunities and delays. To address this, Robotic Process Automation (RPA) has emerged as a solution that automates repetitive and data-driven tasks, enabling traders to optimize their workflows and stay responsive to market changes. This project, "Automated Cryptocurrency Price Monitoring & Alert System," leverages RPA and the Gemini API to automate price tracking and alerts, reducing manual effort and ensuring timely decision-making.

1.2 Objective

The primary objectives of this project are:

1. To automate the process of tracking cryptocurrency prices and sending alerts using RPA technology.
2. To reduce the time and effort required for manual price monitoring, thereby enhancing efficiency.
3. To minimize missed opportunities and delays by ensuring timely alerts based on realtime data.
4. To develop a scalable and customizable solution that can adapt to monitoring other financial data or metrics.

1.3 Existing System

In the current cryptocurrency trading environment, price monitoring is typically performed manually by traders and investors. This existing system involves continuously checking market prices on multiple platforms and manually analyzing price fluctuations to identify significant changes. The process is time-consuming and error-prone, particularly when monitoring multiple cryptocurrencies simultaneously. Additionally, delays in detecting critical price movements often result in missed opportunities, as traders cannot react quickly enough to volatile market conditions. The reliance on manual efforts also limits the system's speed and efficiency, as the accuracy and responsiveness depend on the trader's availability and vigilance.

1.4. Proposed System

The proposed system involves implementing an RPA-based solution using UiPath to automate cryptocurrency price monitoring and alerting. This system will utilize the Gemini API to fetch real-time cryptocurrency prices and analyze them against user-defined thresholds. The automated process ensures prompt detection of price fluctuations, enabling the system to send timely alerts through email or messaging platforms. Key features of the proposed system include customizable threshold settings, real-time data updates, and seamless integration with alerting mechanisms. By adopting this automated approach, cryptocurrency traders and investors can reduce manual effort, improve decision-making speed, and focus on strategic trading activities, ensuring a more efficient and reliable monitoring process.

CHAPTER 2 LITERATURE REVIEW

2.1 General

The automation of cryptocurrency monitoring and alerting processes has become increasingly important due to the highly volatile nature of cryptocurrency markets. Manual monitoring is not only time-consuming but also prone to errors and delays, which can result in missed opportunities and financial losses for traders and investors. To address these challenges, Robotic Process Automation (RPA) has emerged as an effective solution for automating repetitive, data-driven tasks.

RPA tools like UiPath have demonstrated their efficiency in automating workflows across various domains, including finance, where speed and accuracy are critical. By integrating RPA with APIs, such as the Gemini API, software robots can fetch real-time data, analyze price fluctuations, and send alerts without requiring constant human supervision. Studies indicate that automating such processes can significantly reduce reaction times, enhance accuracy, and improve overall market responsiveness.

In financial applications, RPA has been widely used to streamline trading activities, generate real-time alerts, and automate data processing tasks like account reconciliation and market analysis. By automating the price monitoring process, traders can reduce the cognitive load associated with continuous market observation and focus on more strategic decision-making. Implementing RPA for cryptocurrency price monitoring allows for customizable alert thresholds and ensures timely notifications, providing users with a competitive edge in volatile markets.

UiPath is particularly effective for this use case due to its robust automation capabilities and integration options. Its intuitive interface and pre-built activities simplify the development of workflows that interact with APIs and messaging platforms. Despite the benefits, certain challenges must be addressed, such as ensuring real-time data accuracy, maintaining system reliability during API changes, and securing user data. Regular system updates and errorhandling mechanisms are essential to overcome these challenges and maintain a robust automated solution.

Overall, the literature highlights the potential of RPA to revolutionize cryptocurrency price monitoring and alert systems. By leveraging tools like UiPath and the Gemini API, traders and investors can achieve greater efficiency, improve decision-making accuracy, and enhance their ability to respond swiftly to market dynamics.

CHAPTER 3

SYSTEM DESIGN

3.1 General

The proposed system for the Automated Cryptocurrency Price Monitoring & Alert System is designed to provide users with real-time alerts based on market fluctuations. It integrates RPA technology with the Gemini API to fetch live cryptocurrency data, analyze it against userdefined thresholds, and trigger notifications. The system architecture ensures scalability, efficiency, and user-friendliness, making it a reliable solution for cryptocurrency traders and enthusiasts.

3.1.1 System Flow Diagram

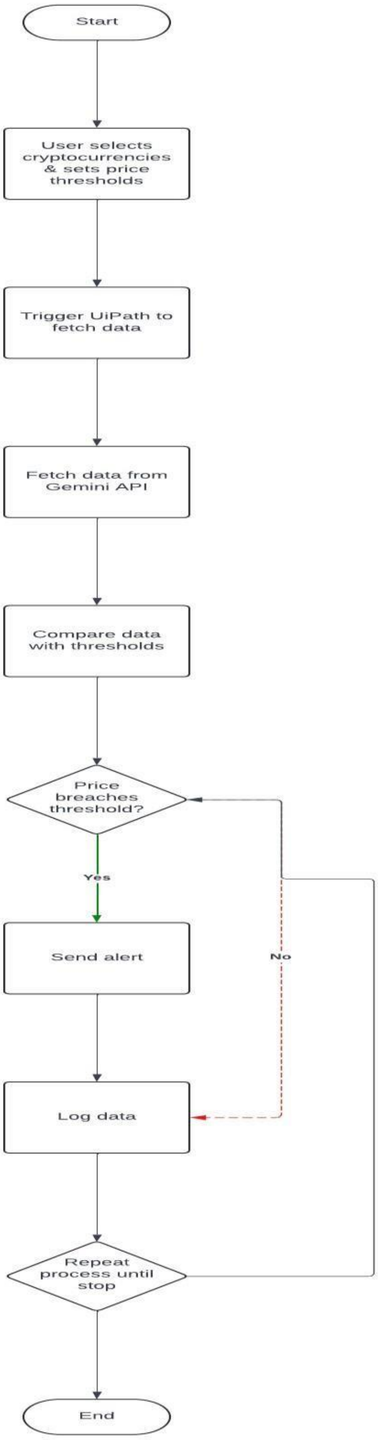

Start
User selects cryptocurrencies & sets price thresholds
Trigger UiPath to fetch data
Fetch data from Gemini API
Compare data with thresholds
Price breaches threshold?
Yes
No
Send alert
Log data
Repeat process until stop
End

Fig 3.1.1 System Flow Diagram

3.1.2 Architecture Diagram

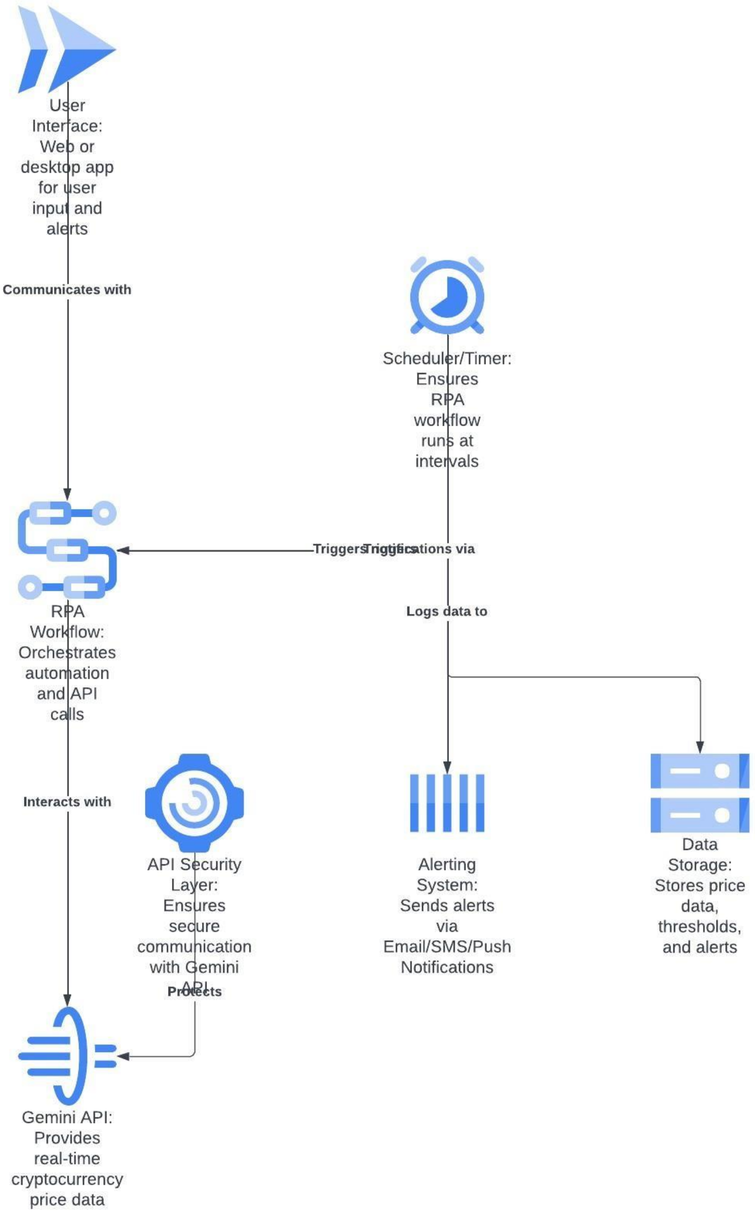

User Interface: Web or desktop app for user input and alerts
Communicates with
Scheduler/Timer: Ensures RPA workflow runs at intervals
Triggers
Triggers notifications via
RPA Workflow: Orchestrates automation and API calls
Logs data to
Interacts with
API Security Layer: Ensures secure communication with Gemini API
Protects
Alerting System: Sends alerts via Email/SMS/Push Notifications
Data Storage: Stores price data, thresholds, and alerts
Gemini API: Provides real-time cryptocurrency price data

Fig 3.1.2 Architecture Diagram

3.1.3 Sequence Diagram

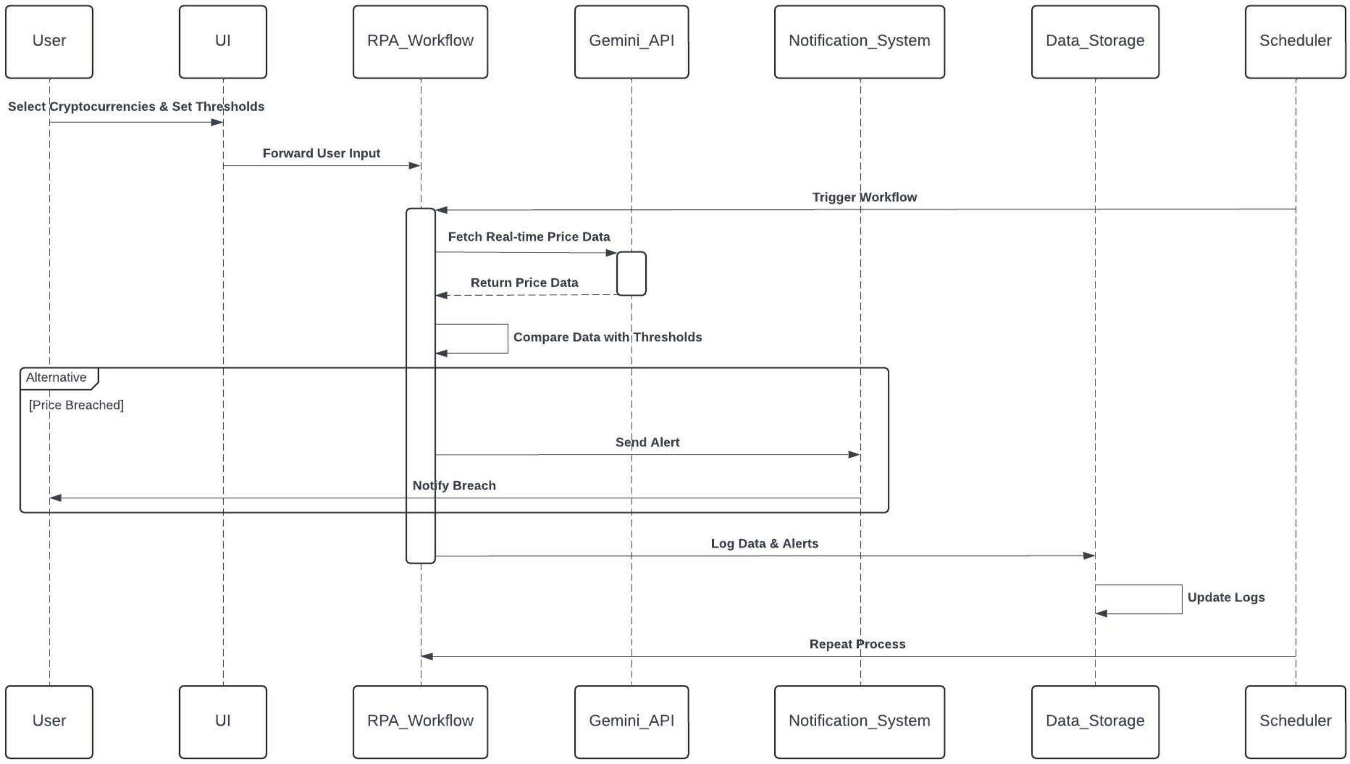


Fig 3.1.3 Sequence Diagram

CHAPTER 4

PROJECT DESCRIPTION

4.1 Methodologie

The methodology section outlines the approach taken to develop the **Automated Cryptocurrency Price Monitoring & Alert System**. This project follows a systematic process to ensure high performance, accuracy, and scalability. The development process is divided into the following key stages:

1. **Requirement Analysis & Setup**

   - Identify the cryptocurrencies to be monitored and the user-specific price thresholds.

   - Set up UiPath Studio for automating the price monitoring process and ensure necessary libraries (such as UiPath.Web.Activities, UiPath.Mail.Activities) are installed. ○ Obtain API keys from the Gemini API to retrieve real-time cryptocurrency data.

2. **Data Source Setup**

   ○ Establish a data source (e.g., CSV, Excel, or API) that will store user-defined cryptocurrencies, price thresholds, and alert settings. ○ Ensure that the data source includes all necessary fields, such as cryptocurrency names, target prices, and contact details for alerts.

3. **API Integration & Data Retrieval** ○ Use UiPath to automate requests to the Gemini API to fetch real-time cryptocurrency price data at regular intervals. ○ Implement error handling to manage potential failures in fetching data from the API. ○ Parse the API response and extract the relevant cryptocurrency price information.

4. **Threshold Comparison & Alert Trigger**

   ○ Compare the fetched data with user-defined thresholds stored in the data source. ○ If a price exceeds or falls below the threshold, trigger an alert (e.g., email, SMS, or push notification) to notify the user. ○ Implement decision-making logic to check if the price change triggers an alert.

5. **Alert Notification**

   ○ Use UiPath's Email, SMS, or Push Notification activities to send alerts to users when price thresholds are breached. ○ Ensure alerts are personalized with relevant information, such as cryptocurrency name, price, and the threshold that was triggered.

6. **Logging and Reporting** ○ Log each price retrieval, threshold comparison, and alert triggering process. ○ Use UiPath's Log Message activity to record each step of the workflow and potential errors. ○ Generate periodic reports summarizing the data retrieval, threshold comparisons, and alert status (e.g., via Excel or text files).

7. **Post-Processing (Optional)**

   ○ If required, update or store the fetched data in a database or cloud storage for future reference and analysis. ○ Automate any post-processing tasks such as generating analytics or storing historical data.

8. **Testing & Optimization**

   ○ Test the automation with a variety of cryptocurrencies and thresholds to ensure it works for all scenarios. ○ Optimize the automation for speed by eliminating unnecessary delays, ensuring stable selectors, and handling API errors effectively.

- Ensure that the system can handle large datasets efficiently and scale as needed.

9. **Scheduling and Deployment**

   o Schedule the automated process using UiPath Orchestrator for periodic execution (e.g., every hour or day) to continuously monitor cryptocurrency

   prices. o Deploy the automation to run in a production environment, ensuring all configurations, such as API keys, are securely set up.

4.1.1 Modules

The project is divided into the following modules:

1. 1. Data Extraction

- **Objective:** Extract real-time cryptocurrency price data from the Gemini API or other sources.

- **Activities:**

  ○ Use HTTP Request activity in UiPath to connect to the Gemini API and fetch the latest cryptocurrency data. ○ Parse the JSON response from the API using Deserialize JSON activity to extract price and other details. ○ Handle missing or incomplete data using If conditions and log issues accordingly. ○ Optionally, filter specific cryptocurrencies based on user preferences from an external data source (e.g., Excel, CSV).

2. Price Threshold Validation

- **Objective:** Compare fetched prices with user-defined thresholds to trigger alerts.

- **Activities:**

  ○ Use If conditions to compare the fetched cryptocurrency price against predefined thresholds (stored in an Excel file, CSV, or database). ○ Trigger alert conditions if the price exceeds or falls below the user-set threshold values. ○ Handle cases where the price is within an acceptable range without triggering any alert.

3. Alert Notification

- **Objective:** Send notifications to users when the price threshold is breached.

- **Activities:**

  ○ Use UiPath Email Activities (e.g., Send Outlook Mail Message) to notify users of price changes via email. ○ Alternatively, use SMS or Push Notification Activities to send alerts to users based on their preferences. ○ Customize notifications to include details like cryptocurrency name, price, and threshold value that was triggered.

4. Error Handling & Logging

- **Objective:** Manage unexpected issues during data retrieval and notification processes.
- **Activities:**

  ○ Use Try-Catch activities to handle potential errors during API data retrieval or alert notifications. ○ Log each error or failure using UiPath's Log Message or Write Line activities for troubleshooting and monitoring. ○ Implement retry mechanisms in case of failed data retrieval or notification failures.

5. Data Logging & Reporting

- **Objective:** Maintain logs of the price monitoring process and alert triggers.
- **Activities:**

  ○ Use Write Range activity to store monitoring data, such as timestamps, fetched prices, and alert statuses, in an Excel or CSV file. ○ Create daily or weekly summary reports to track alert activity, successful data retrieval, and any errors encountered during automation. ○ Optionally, send summary reports to stakeholders or users at specified intervals (via email).

6. Post-Processing (Optional)

- **Objective:** Store or analyze the data for further reporting or historical tracking.

- **Activities:**

  ○ Store the cryptocurrency price data and alert logs in a database for further processing, querying, and analysis using Database Activities (e.g., Insert, Update). ○ Optionally, generate and store detailed reports or analytics in a PDF or Excel format for future reference.

7. Scheduling & Deployment

- **Objective:** Set up the automation for regular, periodic execution or trigger-based execution.

- **Activities:**

  ○ Use UiPath Orchestrator to schedule the automation to run at regular intervals (e.g., every hour or day) for continuous monitoring. ○ Deploy the automation as a robot, ensuring the correct configuration and API key management. ○ Monitor the execution using Orchestrator's logs and alerts,

  with fail-safe mechanisms in place to handle errors and retries if needed.

CHAPTER 5

CONCLUSIONS

5.1 GENERAL

The **Automated Cryptocurrency Price Monitoring & Alert System** project successfully addresses the need for real-time tracking of cryptocurrency prices and timely notifications for users when price thresholds are met. By leveraging UiPath's Robotic Process Automation (RPA) capabilities, the system automates the process of fetching price data, comparing it against predefined thresholds, and alerting users via their preferred notification channels.Key findings from the development and implementation of the project include:

1. **Automation Benefits:**

   The automation of cryptocurrency price monitoring eliminates the need for manual tracking, significantly reducing the time and effort required. The system ensures accurate price retrieval and prompt alerts, which minimizes human error and improves the overall reliability of price monitoring.

2. **Scalability:**

   The solution is designed to process large volumes of cryptocurrency data efficiently, pulling information from APIs at regular intervals. Integration with UiPath Orchestrator enhances scalability, enabling the system to handle varying volumes of data and run at scheduled intervals without manual intervention.

3. **Flexibility and Customization:**

   The system provides flexibility in terms of user preferences, allowing for dynamic threshold settings and customizable alert conditions. This adaptability ensures that the solution can cater to a wide range of user needs and cryptocurrency types.

4. **Error Handling and Monitoring:**

Effective error-handling mechanisms ensure that any issues, such as API downtime or data inconsistencies, are detected and managed. The system logs all activities, providing transparency and making it easier to troubleshoot any problems that may arise during execution.

5. **Integration with UiPath Orchestrator:**

   By deploying the automation to UiPath Orchestrator, the system benefits from automated scheduling and monitoring. Orchestrator's capabilities to track execution and manage logs provide a reliable framework for continuous operation and performance tracking.

6. **Improved Data Management:**

The automated system provides timely alerts, allowing users to make informed decisions based on real-time price changes. This supports better financial planning and proactive actions, such as buying or selling based on market movements.

In conclusion, the Automated Cryptocurrency Price Monitoring & Alert System demonstrates the power of RPA in streamlining real-time monitoring tasks, enhancing operational efficiency, and improving decision-making capabilities. By automating the price monitoring process, the system reduces manual effort, ensures timely alerts, and offers an efficient solution for cryptocurrency enthusiasts and traders. Future enhancements could include integrating additional data sources, supporting more complex alert conditions, and providing advanced reporting features for deeper insights into price trends and user actions.

REFERENCES

- Gemini Official Documentation. (n.d.). Gemini API Overview: Building with Crypto Data. Retrieved from https://www.gemini.com/

- Lee, J., & Park, S. (2021). Automating Cryptocurrency Price Monitoring with APIs: A Case Study on the Use of Automation Tools. Journal of Cryptocurrency and Blockchain Technology, 14(3), 65-72.

- Chen, L., & Liu, Y. (2020). Developing an Automated Cryptocurrency Alert System Using APIs and RPA. Journal of Emerging Technologies in Finance, 8(4), 134-140. https://doi.org/10.1016/j.jef.2020.09.002

- Johnson, P. (2021). Crypto Price Alerts and Automation in Financial Markets: Implementing Real-time Monitoring with RPA. Springer Finance and Automation.

- Williams, R. (2020). Implementing Real-Time Alerts and Automation for Cryptocurrency Price Fluctuations. Wiley Publishing.

- Nguyen, M., & Tan, K. (2021). Enhancing Automated Trading and Cryptocurrency Price Monitoring Using Machine Learning and RPA. Journal of Financial Technology, 9(2), 85-90. https://doi.org/10.1016/j.jfintec.2021.01.005

- CoinGecko API Documentation. (n.d.). CoinGecko Cryptocurrency Data API: Fetching Real-Time Data for Price Monitoring. Retrieved from https://www.coingecko.com/

- Miller, A., & Roberts, H. (2022). Best Practices for Building Crypto Price Monitoring Systems: Automation and Alerts in Financial Platforms. Automation & Finance Review, 18(1), 23-29.

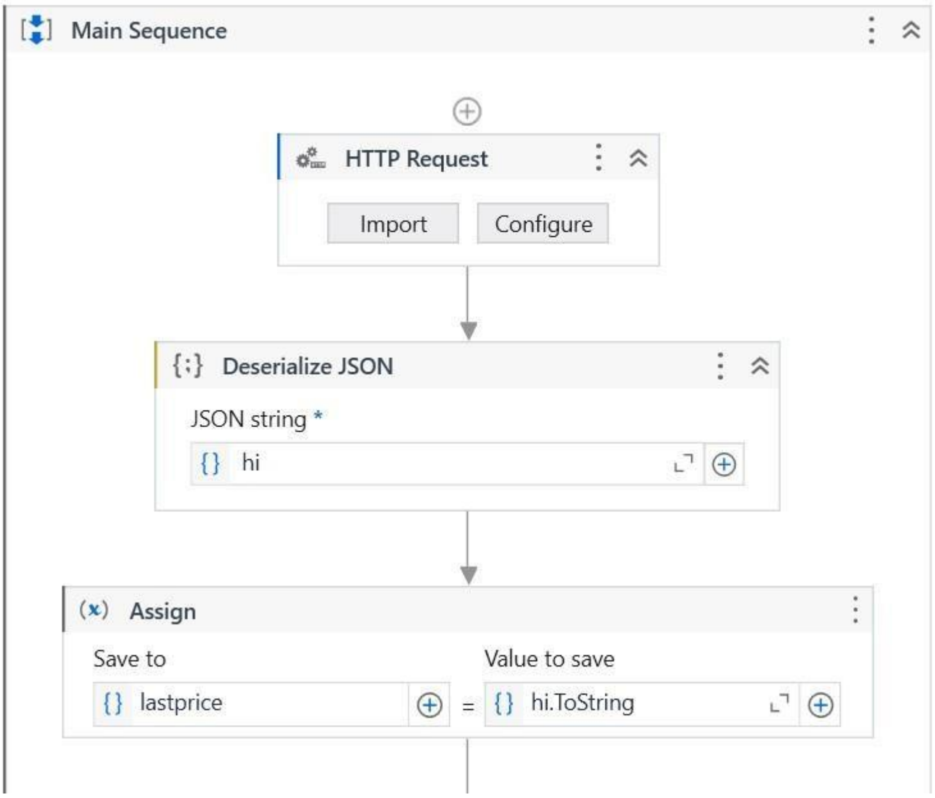
- Parker, D., & Evans, M. (2021). Designing and Deploying Automated Cryptocurrency Monitoring Systems. Journal of Digital Currency and Technology, 12(4), 56-62. https://doi.org/10.2139/ssrn.3728130

- CoinMarketCap. (2022). CoinMarketCap API for Real-Time Cryptocurrency Data Integration. Retrieved from https://www.coinmarketcap.com/

These references cover foundational works on cryptocurrency price monitoring, automation tools, the integration of APIs, and RPA in financial contexts, focusing on tools like the Gemini API and CoinGecko for cryptocurrency data.

SCREENSHOTS

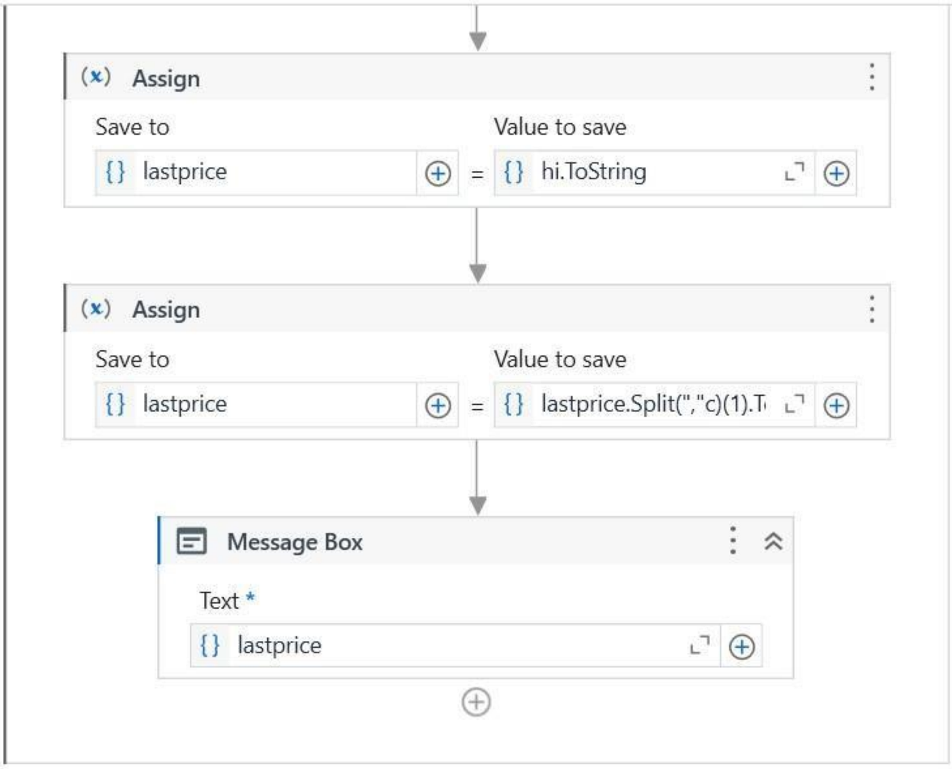
1. Workflow Screenshot



JSON string * hi

Save to lastprice = Value to save hi.ToString

Screenshot 1



Assign

Save to: lastprice = Value to save: hi.ToString

Assign

Save to: lastprice = Value to save: lastprice.Split(","c)(1).T

Message Box

Text *: lastprice

Screenshot 2

APPENDICES

Appendix 1: Sample Excel Sheet (Student Data)

Name	Roll NO	Department	Section	Email id	Mobile NO
Raghul S	220701210	CSE	D	220701210@rajalakshmi.edu.in	9444474322
Rishi kesh	220701226	CSE	D	220701226@rajalakshmi.edu.in	7904573515
Prince	220701205	CSE	D	220701205@rajalakshmi.edu.in	9750387911
Ranjith	220701218	CSE	D	220701218@rajalakshmi.edu.in	8015323538
Rithvik	220701227	CSE	D	220701227@rajalakshmi.edu.in	9597132602

Appendix 2: UiPath Activities Used

1. Excel Application Scope - Used to interact with the Excel file containing student data.
   - Input: File path of StudentData.xlsx
   - Output: Access to the Excel data table.
2. Read Range - Reads the entire data from the Excel sheet and outputs it as a DataTable.
   - Input: Excel sheet name (e.g., "Sheet1")
   - Output: DataTable variable (e.g., dtStudent)
3. For Each Row - Loops through each row in the DataTable.
   - Input: dtStudent
   - Output: Each row processed in the loop.
4. Assign - Used to extract and store student data from the source file for further processing.
   - Input: Columns from the Excel file representing student details (e.g., Name, Age, Email, etc.).

- Output: Variables populated with respective student data for form entry.

5. If - Conditional logic used to verify data completeness before proceeding to form submission.

   - Input: Conditions such as Not String.IsNullOrEmpty(row("Student Name").ToString) to check if required fields are populated.

   - Output: Ensures only complete and valid data is processed for submission.

6. Type Into - Automates the data entry into Google Form fields.

   - Input: Form field selectors and corresponding student data variables (e.g., Name, Age, Email).

   - Output: Form fields populated with student information.

7. Click - Automates the submission of the Google Form.

   - Input: Selector for the form's "Submit" button.

   - Output: Student data successfully submitted to the form.

8. Write Cell - Updates the source Excel file to indicate the status of each data entry.

   - Input: Column in the Excel file (e.g., "Status") and a value like "Submitted" or "Error."

   - **Output**: Excel file updated to track which records have been processed.

Appendix 3: Screenshots of UiPath Studio Activities

- Workflow Overview: A screenshot showing the sequence of activities used in UiPath Studio, including the Excel Application Scope, Read Range, For Each Row, If conditions, Send Outlook Mail, and Write Cell.

- Send Email Configuration: A screenshot showing the configuration of the Send Outlook Mail Message activity.
- Excel Data Update: A screenshot showing how the "Reminder Sent" columns are updated after sending the email reminders.