CONTRACT EXPIRY NOTIFICATION BOT

A PROJECT BY REPORT

Submitted by

SHANMUGA DIVYA K (220701261)

in partial fulfilment for the course

OAI1903 – INTRODUCTION TO ROBOTIC PROCESS AUTOMATION

for the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR

THANDALAM

CHENNAI – 602 105

NOVEMBER 2024

RAJALAKSHMI ENGINEERING COLLEGE

CHENNAI - 602105

BONAFIDE CERTIFICATE

Certified that this project report "**CONTRACT EXPIRY NOTIFICATION BOT**" is the Bonafide work of "**SHANMUGA DIVYA K (220701261)**" who Carried out the project work for the subject OAI1903 – Introduction to Robotic Process Automation under my supervision

Dr. N. Durai Murugan

SUPERVISIOR

Associate 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 _____.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of ourlife 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 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 guides,* ***Ms. Roxanna Samuel,M.E.,*** *Assistant Professor (SG),* ***Ms. U.Farjana****, M.E., Assistant Professor and* ***Ms. S.Vinothini****,* ***M.E****., 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,* ***Dr. P.Revathy, M.E., Ph.D.,*** *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.*

- SHANMUGA DIVYA K (220701261)

ABSTRACT

The Contract Expiry Notification System harness Robotic Process Automation (RPA) to streamline contract management by automating the tracking of contract expiration dates. This intelligent bot continuously monitors active contracts, generating timely notifications when key deadlines approach. By integrating seamlessly with existing contract management platforms, the system eliminates manual oversight and reduces the risk of lapses that can lead to service disruptions or compliance issues

Designed to enhance operational efficiency, the bot provides customizable alerts tailored to specific contracts, allowing organizations to prioritize renewals and negotiations based on urgency and importance. This proactive notification system ensures that stakeholders are informed well in advance, facilitating timely actions and strategic decision-making.

Moreover, the RPA framework allows for scalability, enabling the system to adapt to varying contract volumes without additional manual effort. By automating routine tasks associated with contract monitoring, organizations can redirect resources to more strategic initiatives, thus optimizing their overall contract management processes.

In summary, the Contract Expiry Notification System exemplifies the transformative potential of RPA in contract management, providing organizations with the tools to enhance compliance, improve vendor relationships, and ensure continuous operational flow. By leveraging automation, businesses can proactively manage their contracts, safeguarding their interests and driving long-term success.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

ABBREVATION	ACCRONYM
RPA	Robotic Process Automation
AI	Artificial Intelligence

CHAPTER 1

INTRODUCTION

1.1 General :

Managing contract renewals is a critical process for businesses to ensure seamless operations and compliance. However, tracking contract expirations manually can be prone to errors, leading to missed deadlines, financial penalties, or operational disruptions. In today's fast-paced environment, automated solutions are increasingly being adopted to streamline such tasks efficiently.

1.2 OBJECTIVE:

The objectives of the Contract Expiry Notification Bot are:

1. Automate Monitoring: Automatically track and monitor contract expiration dates to eliminate the need for manual tracking.

2. Timely Notifications: Send proactive alerts to stakeholders well in advance of contract expirations, ensuring adequate time for action.

3. Reduce Errors: Minimize human errors associated with manual tracking and reminder system.

4. Enhance Decision-Making: Provide stakeholders with timely information to support decisions on contract renewals, renegotiations, or terminations.

1.3 Existing System

Currently, most organizations rely on manual tracking systems, such as spreadsheets, emails, or basic reminders, to manage contract expiration dates. These systems often lack real-time notifications, scalability, and the ability to integrate with existing enterprise solutions. Human error and inefficiencies in the current system can lead to costly consequences, such as lapsed contracts, missed renewal opportunities, and compliance breaches.

1.4 Proposed System

The Contract Expiry Notification Bot offers an intelligent, automated solution to address the limitations of existing systems. This bot integrates with contract databases or management software to monitor expiration dates in real time. It sends timely notifications via email, SMS, or instant messaging platforms to relevant stakeholders. The solution includes customizable alerts, integration capabilities, and a user-friendly interface to ensure seamless adoption. By automating this critical process, the bot enhances reliability, reduces manual workload, and minimizes risks associated with missed contract renewals.

The proposed solution for the Contract Expiry Notification Bot includes the following key features and functionalities

1. Automated Tracking:
Continuously monitor contract expiration dates by integrating with contract management databases or systems.

2. Proactive Alerts:
Generate and send timely notifications to stakeholders via preferred communication channels, such as email, SMS, or instant messaging platforms.

3. Customizable Notifications:
Allow users to configure alert settings, such as notification frequency, lead time before expiration, and recipient groups.

4. Data Security:
Implement robust security measures to protect sensitive contract information during tracking and notification processes.

5.Cost-Efficiency:
Reduce resource expenditures by automating manual processes and minimizing risks associated with missed deadlines.

CHAPTER 2

LITERATURE REVIEW

2.1 : General

The field of contract management has seen significant advancements with the integration of automation and digital tools, addressing the challenges of manual tracking and communication. Literature on contract management systems and notification tools highlights the increasing reliance on technology to streamline processes, improve efficiency, and mitigate risks associated with contract expiration and renewal.

Key studies emphasize the role of automated systems in providing real-time alerts, reducing human errors, and supporting compliance with contractual obligations. These systems are often embedded within larger enterprise resource planning (ERP) or contract lifecycle management (CLM) software, showcasing their adaptability and utility across industries.

The concept of contract expiry notification bots is closely aligned with the broader trends of process automation and artificial intelligence (AI). Such bots are designed to not only track dates but also provide insights, analytics, and recommendations for decision-making. Academic and industry research acknowledges their role in enhancing accountability, ensuring operational continuity, and fostering a proactive approach to contract management.

The literature also identifies challenges in implementing such systems, such as data integration, user adoption, and ensuring security and compliance in handling sensitive contract information. However, advancements in cloud computing, AI, and communication technologies are continually addressing these barriers, making contract expiry notification bots a reliable and scalable solution for modern organizations.

CHAPTER 3

SYSTEM DESIGN

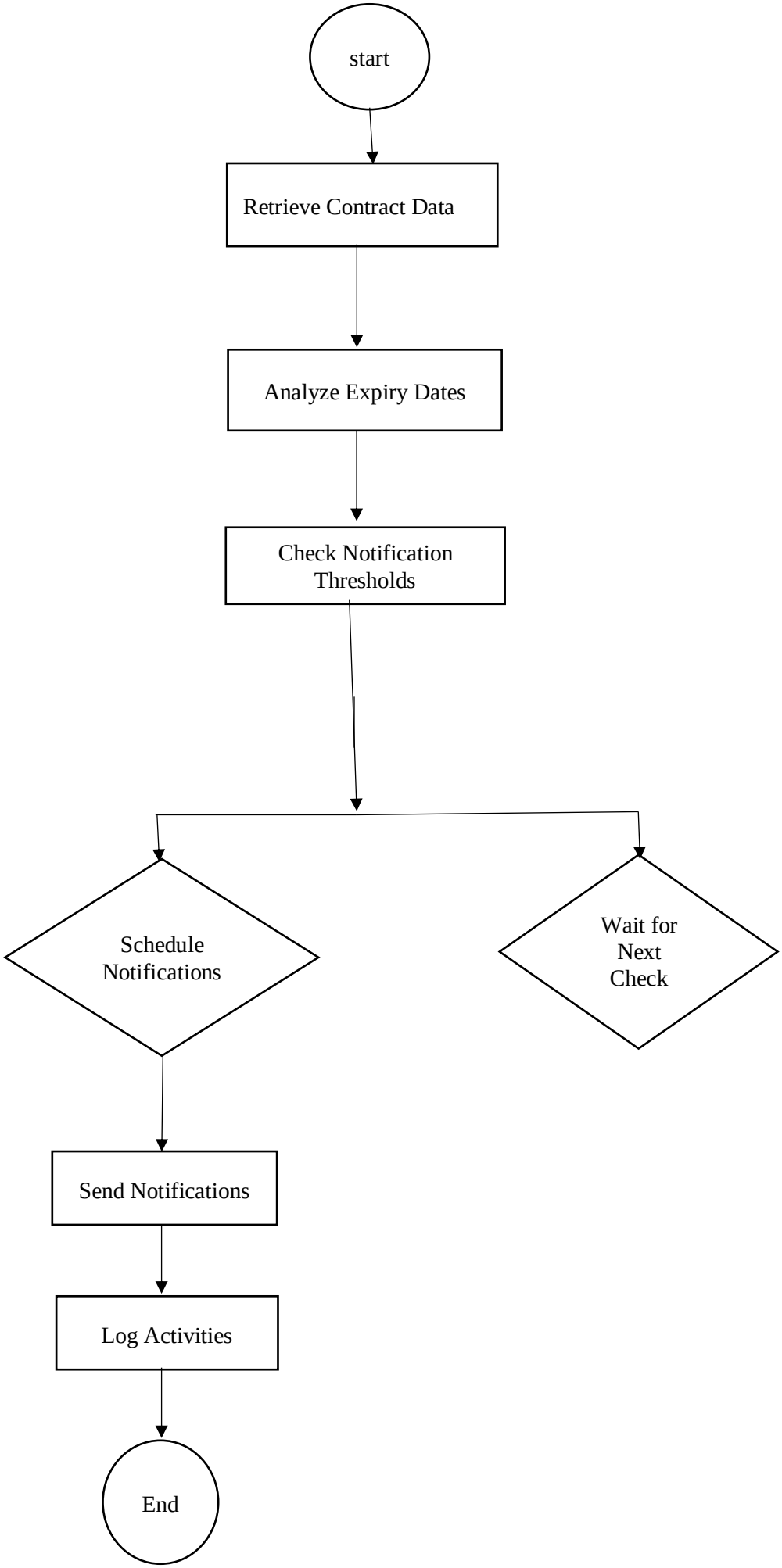
3.1 General

The system design of the Contract Expiry Notification Bot aims to provide a robust framework for automating the tracking, alerting, and reporting of contract expiration events. The design incorporates three key components: data collection and monitoring, notification generation, and user interface interaction. These components ensure that the system is efficient, secure, and scalable while maintaining a user-friendly experience.
The system operates by integrating with a contract database or management system to retrieve relevant data, analyzing expiration timelines, and sending proactive alerts via multiple communication channels. The design also includes options for customization, such as setting thresholds for notifications and preferred communication methods

3.1.1 System Flow Diagram:

Description:
The system flow diagram illustrates the high-level workflow of the Contract Expiry Notification Bot. It includes the following steps:

1. Input: The system retrieves contract details (e.g., contract ID, start/end dates, and stakeholder information).
2. Processing: The bot monitors expiration dates, analyzes thresholds, and schedules notifications.
3. Output: Notifications are sent to stakeholders, and logs are updated


start
Retrieve Contract Data
Analyze Expiry Dates
Check Notification Thresholds
Schedule Notifications
Wait for Next Check
Send Notifications
Log Activities
End

3.1.2 Architecture Diagram:

Description:

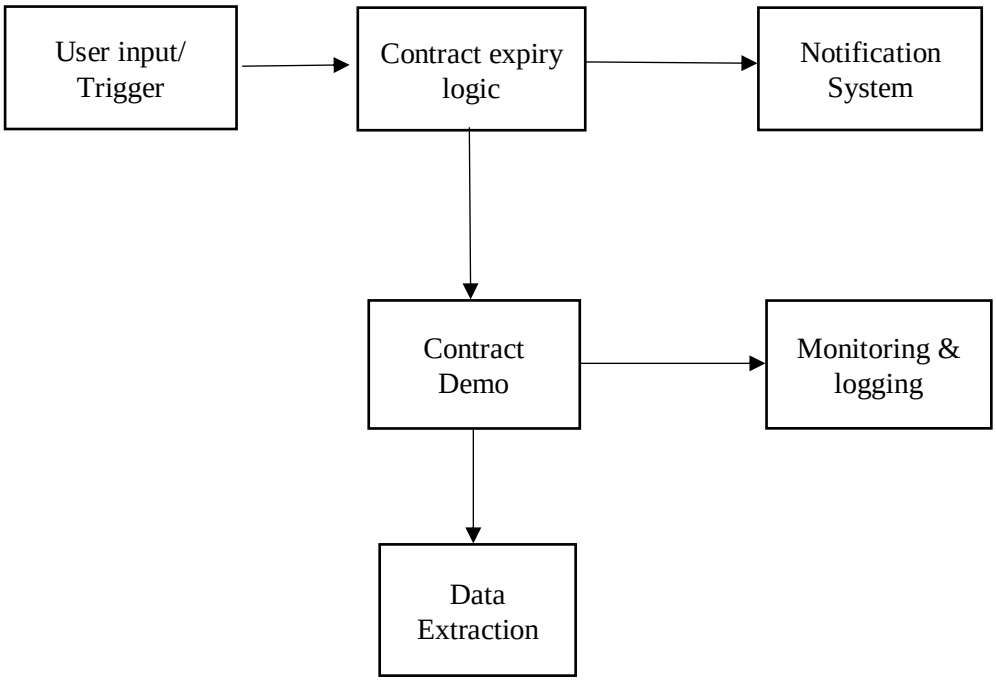
The system architecture defines the technical components and their interactions. It includes the following layers:

1. **Presentation Layer**:
    - User dashboard for managing contracts and notification settings.
    - Accessible via web or mobile interfaces.

2. **Application Layer**:
    - Core logic for monitoring expiration dates and generating alerts.
    - Notification engine to send alerts via email, SMS, or instant messaging.

3. **Data Layer**:
    - Database to store contract details, user preferences, and notification logs.
    - Integration with existing contract management systems or external APIs.

4. **Communication Layer**:
    - APIs to integrate with third-party communication platforms (e.g., Twilio for SMS, Slack for messaging).



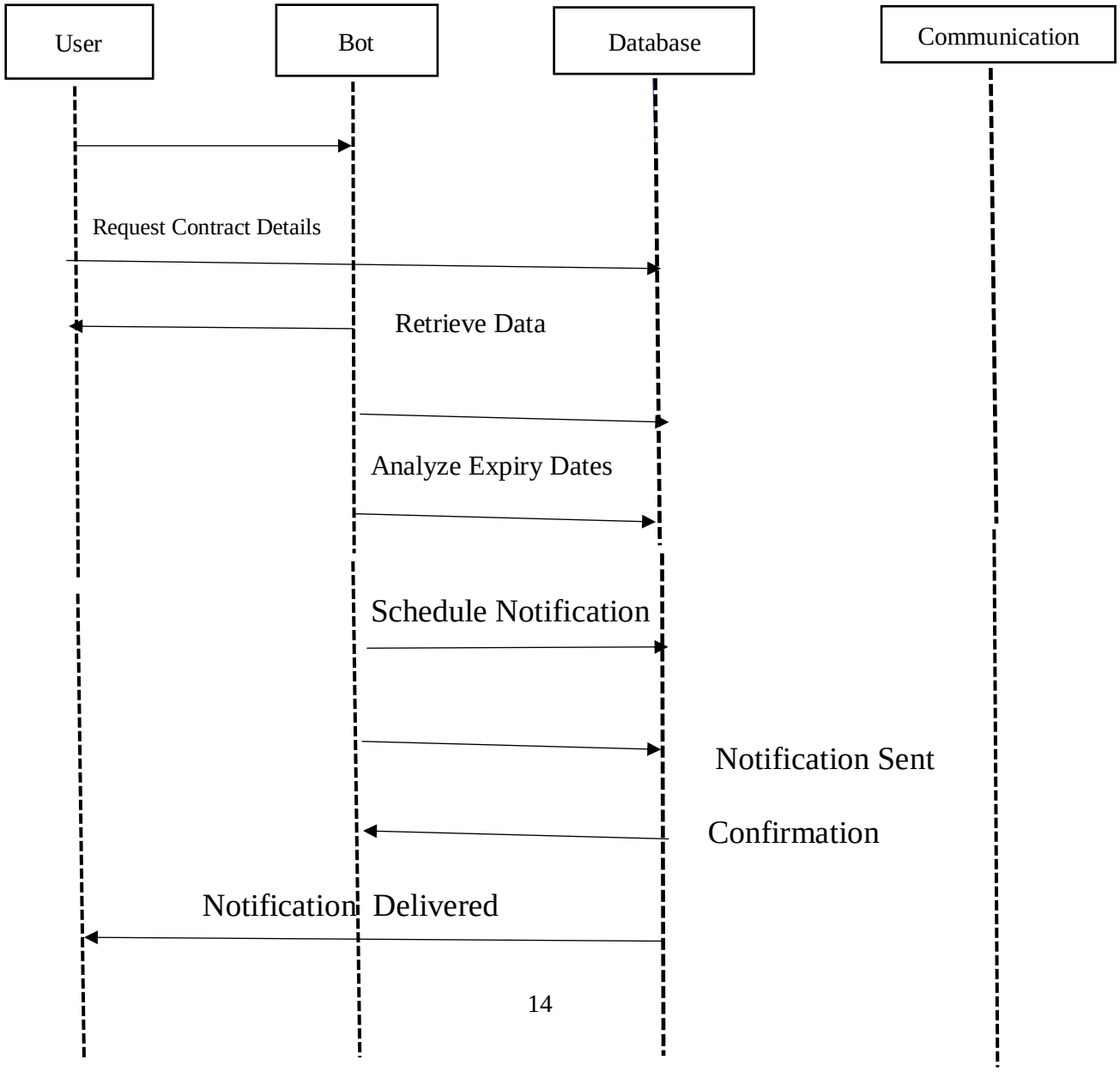
3.1.3 Sequence Diagram

Description:

The sequence diagram shows the interaction between system components during the notification process.

Actors:

- User (Stakeholder or Administrator)
- Contract Management Database
- Notification Bot
- Communication Channel



CHAPTER 4

PROJECT DESCRIPTION

4.1 Methodology

The methodology for the Contract Expiry Notification Bot in RPA involves several key phases: First, it begins with requirement analysis and design, where the business needs and data sources are identified, and the RPA tool is chosen. Next, in the data extraction and integration phase, the bot retrieves contract details from structured and unstructured sources like databases and documents. Then, during the expiry date calculation phase, the bot compares contract end dates with the current date, identifying contracts approaching expiration. The notification delivery phase follows, where the bot sends timely email and SMS alerts to stakeholders about expiring contracts. Finally, the monitoring and logging phase ensures bot performance is tracked, errors are logged, and issues are resolved, with reporting and exception handling ensuring smooth, continuous operation.

Here's the Contract Expiry Notification Bot methodology outlined point by point:

1. Requirement Analysis and Design: Identify business requirements, define stakeholders, determine data sources (databases, documents, etc.), and select the RPA tool for automation.

2. Data Extraction and Integration: Extract contract data from various sources, including databases, document management systems, or unstructured formats (using OCR for scanned files), and ensure data integrity.

3. Expiry Date Calculation: Compare contract expiry dates with the current date to identify contracts that are nearing expiration, applying logic for exclusions (renewals, non-active contracts).

4. Notification Delivery: Send automated notifications via email or SMS to relevant stakeholders based on contract details and predefined templates, with reminders at specified intervals before the expiry date.

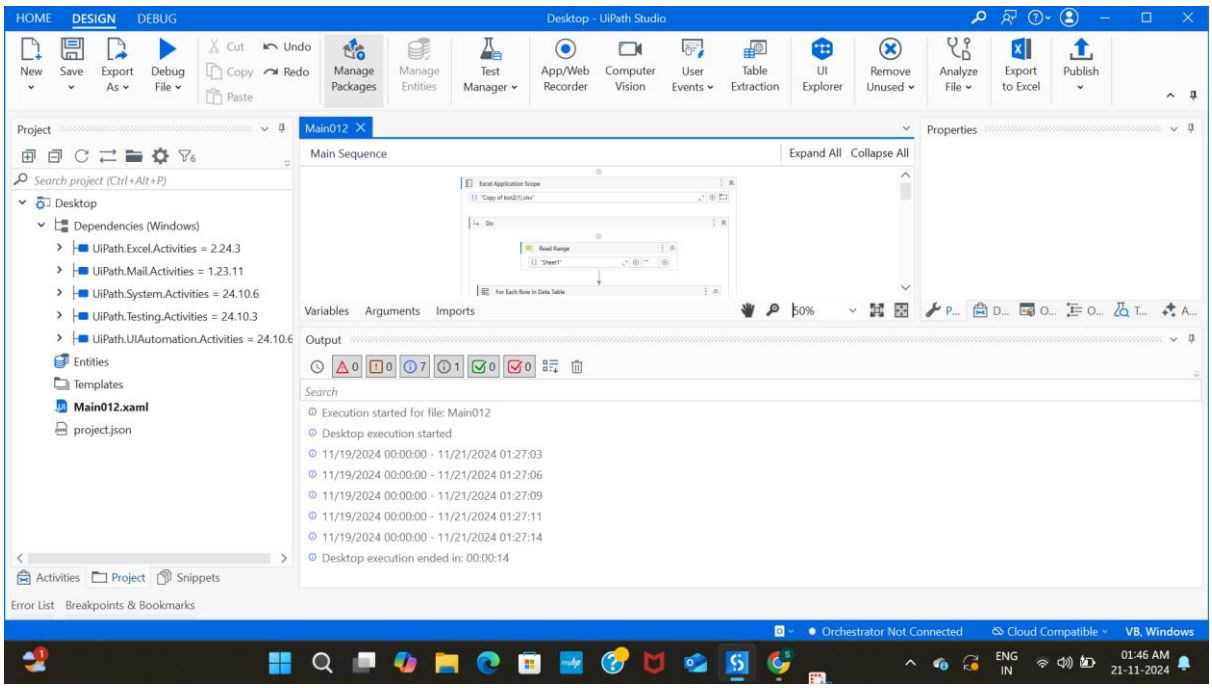
5. Monitoring, Logging, and Exception Handling: Track bot performance, log activities, handle errors (via retries or manual intervention)

4.1.1 Modules

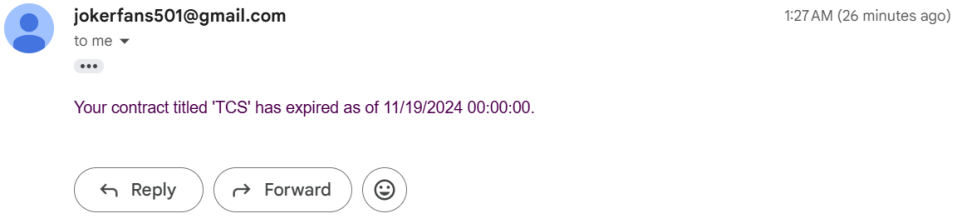
1. **Data Extraction Module**
    - **Purpose**: Extract contract information from multiple data sources such as databases, documents, and cloud storage.
    - **Implementation**: The bot uses RPA tools to connect to databases (SQL, NoSQL), and document management systems (e.g., SharePoint) to pull structured contract data. OCR (Optical Character Recognition) is implemented for extracting data from unstructured formats like scanned PDFs or images.
2. **Expiry Calculation and Analysis Module**
    - **Purpose**: Identify contracts that are nearing their expiry dates and require notifications.
    - **Implementation**: The bot compares the current date to the expiry date field in the extracted contract data, flagging contracts that meet the defined threshold (e.g., 30 days before expiry). Custom logic is used to handle exceptions, such as contracts that are renewed or extended.
3. **Notification Management Module**
    - **Purpose**: Automatically send notifications (email/SMS) to relevant stakeholders about expiring contracts.
    - **Implementation**: The bot uses APIs to send customized email notifications (via SMTP or email services like Gmail or Outlook) and SMS alerts (via SMS gateways like Twilio), with dynamic templates that include contract details. Reminders are scheduled to be sent at set intervals (e.g., 30 days, 7 days before expiry).
4. **Monitoring and Logging Module**
    - **Purpose**: Monitor bot activities and log events for troubleshooting and performance tracking.
    - **Implementation**: The bot logs all key activities, such as data extraction, notification sending, and any errors encountered, into a centralized log system (e.g., UiPath Orchestrator or a custom database). The logs provide insights into the bot's health and performance metrics.
5. **Exception Handling and Resilience Module**
    - **Purpose**: Ensure smooth operation of the bot by handling errors and enabling retries when failures occur.
    - **Implementation**: The bot uses built-in error handling mechanisms in the RPA tool (e.g., try-catch blocks, retry workflows) to manage issues like missing data, failed notifications, or temporary connection issues, automatically retrying actions or notifying administrators when manual intervention is needed.

CHAPTER 5

OUTPUT SCREENSHOT



Contract Title	Expiration Date	Email Address
Creation of AI	19-11-2024	jokerfans501@gmail.com
TCS	19-11-2024	shanmugadivya7@gmail.com
infosys	19-11-2024	manishakarikalan@gmail.com
Hcl	19-11-2024	ay.karikalan@gmail.com
wipro	19-11-2024	ananthikarikalan@rediffmail.com



CHAPTER 6

CONCLUSION

Managing contract expirations is a critical task for organizations, yet it is often prone to errors and inefficiencies when handled manually. The implementation of a contract expiry notification bot using Robotic Process Automation (RPA) revolutionizes this process, offering a seamless and reliable approach to managing contract timelines. By automating the identification of contracts nearing their expiration dates and promptly notifying relevant stakeholders, businesses can ensure that no contract deadlines are overlooked, reducing the risk of penalties or missed renewals.

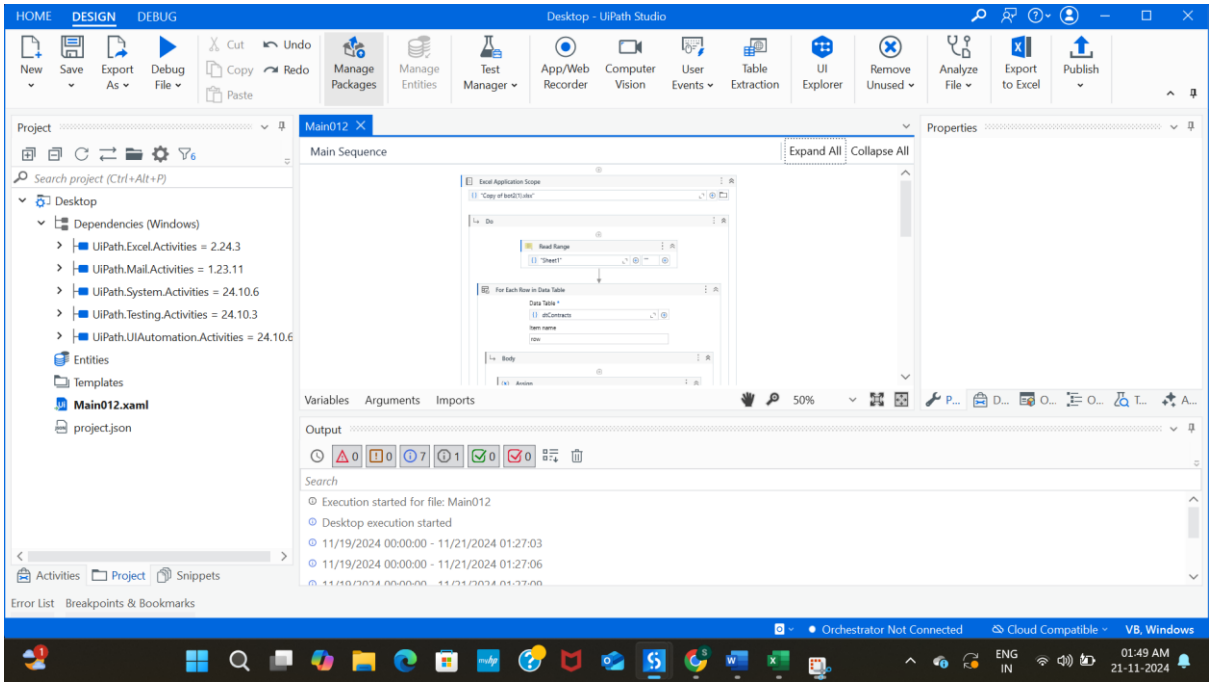
This automation solution not only improves operational efficiency but also enhances compliance with contractual and regulatory obligations. With RPA bots, organizations can process large volumes of contracts swiftly and accurately, eliminating the need for manual tracking or intervention. The bot's ability to integrate with enterprise systems such as email platforms, CRM, and contract management tools ensures seamless execution, further improving workflow efficiency.

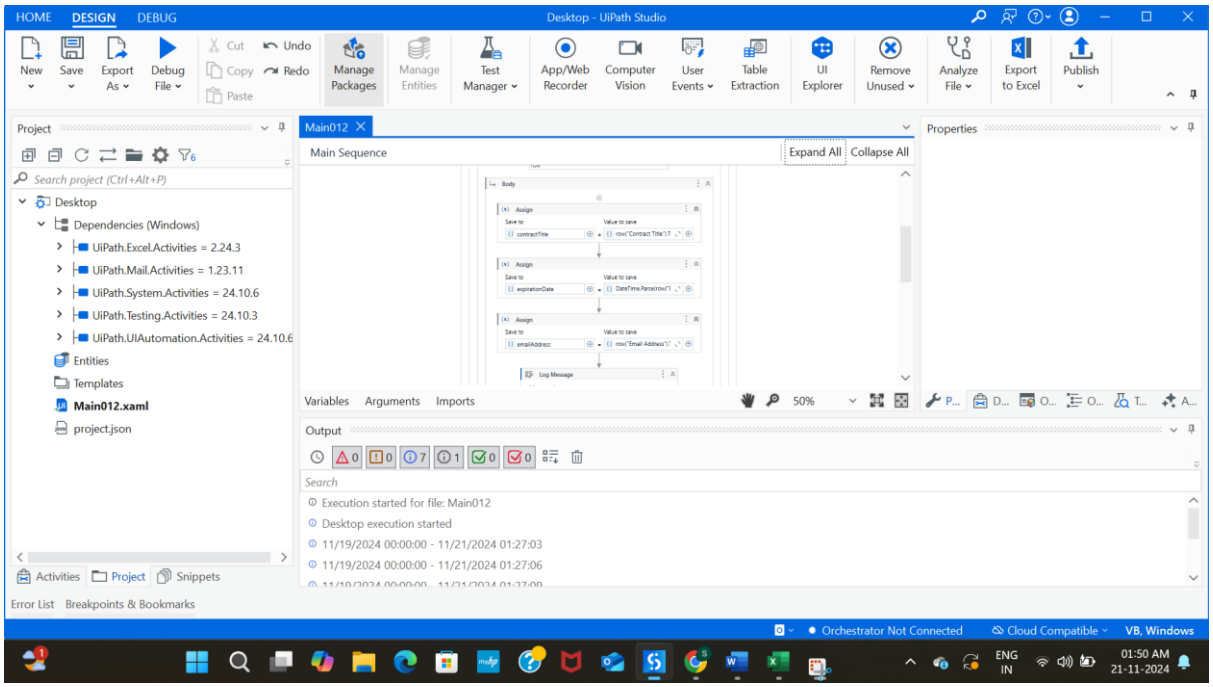
Moreover, by enabling timely reminders and updates, the bot fosters stronger communication between internal teams and external stakeholders. This proactive approach builds trust and credibility, as all parties are kept informed of upcoming deadlines and necessary actions. The scalability of RPA also ensures that the solution can grow with the organization, managing increasing contract volumes without additional manual effort.

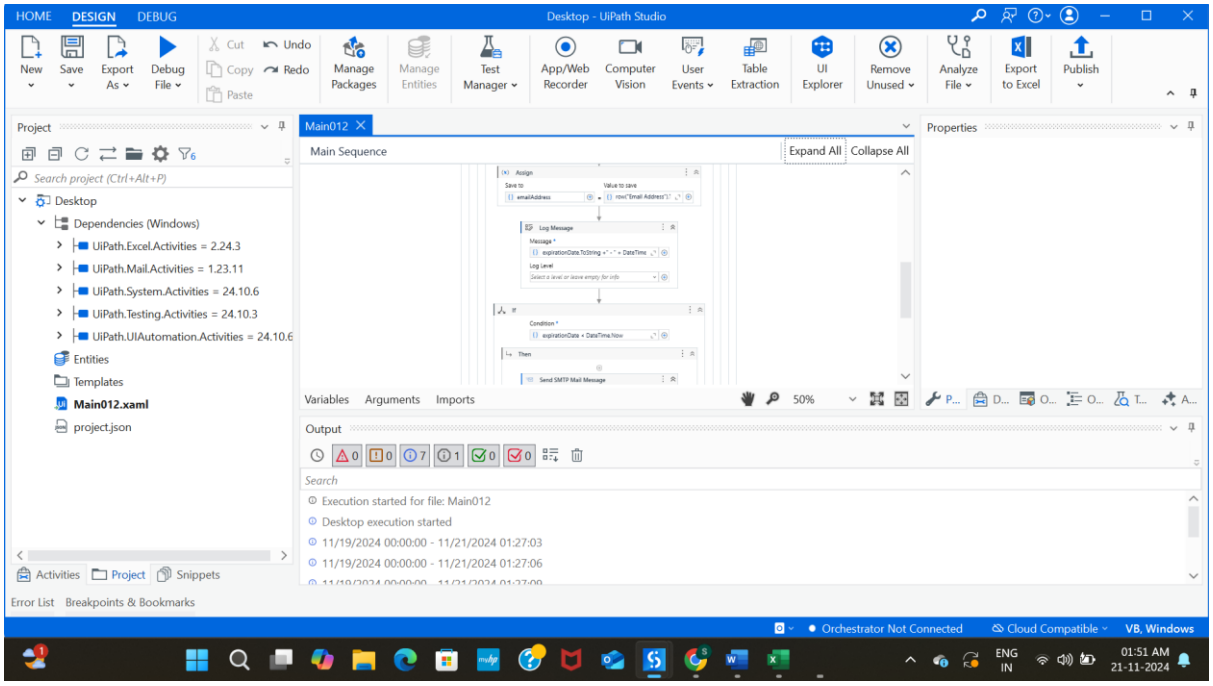
In conclusion, a contract expiry notification bot powered by RPA is a transformative tool for modern businesses. It mitigates risks, reduces operational costs, and provides organizations with a competitive edge through enhanced accuracy and efficiency. By adopting this technology, companies can focus on strategic decision-making while leaving routine, time-sensitive tasks to automation, ultimately driving better business outcomes.

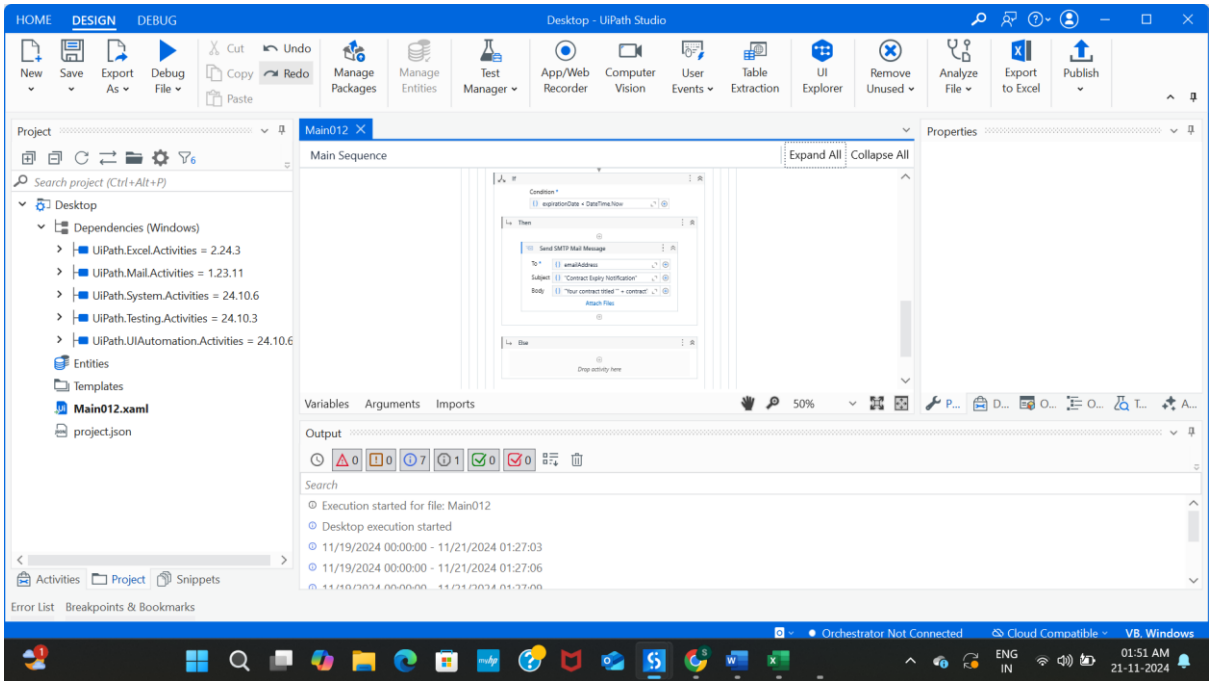
APPENDIX

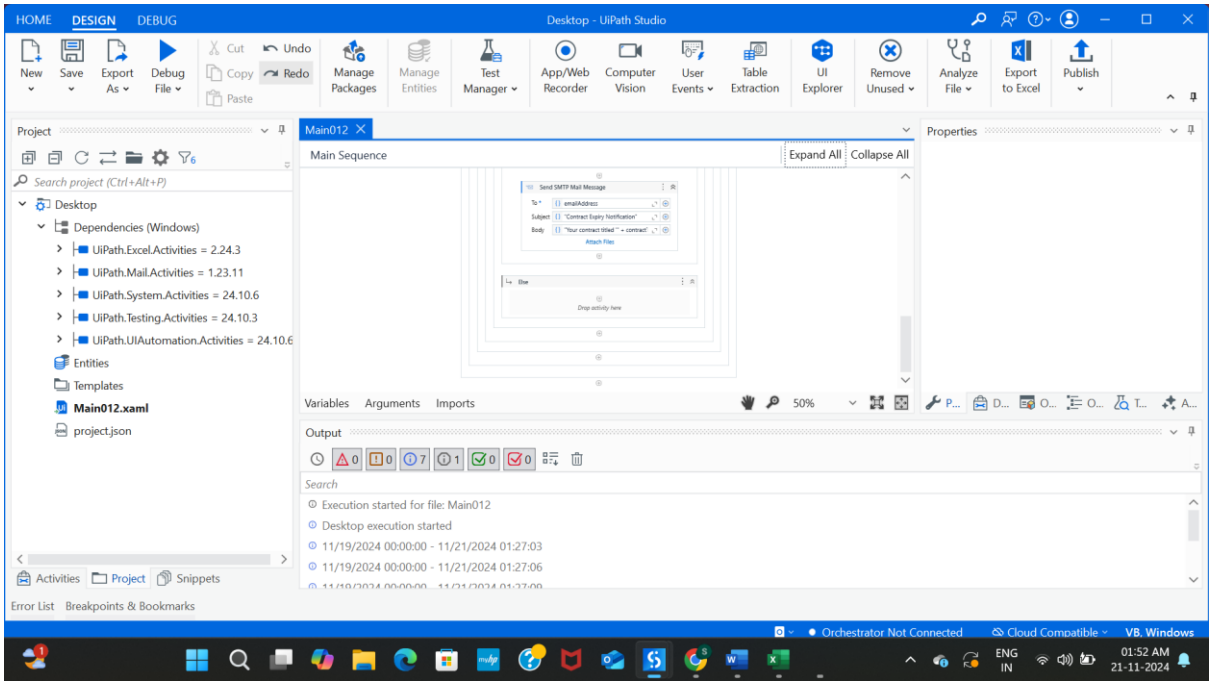
PROCESS WORKFLOW






HOME
DESIGN
DEBUG
Desktop - UiPath Studio
New
Save
Export As
Debug File
Cut
Copy
Paste
Undo
Redo
Manage Packages
Manage Entities
Test Manager
App/Web Recorder
Computer Vision
User Events
Table Extraction
UI Explorer
Remove Unused
Analyze File
Export to Excel
Publish
Project
Search project (Ctrl+Alt+P)
Desktop
Dependencies (Windows)
UiPath.Excel.Activities = 2.24.3
UiPath.Mail.Activities = 1.23.11
UiPath.System.Activities = 24.10.6
UiPath.Testing.Activities = 24.10.3
UiPath.UIAutomation.Activities = 24.10.6
Entities
Templates
Main012.xaml
project.json
Main012
Main Sequence
Expand All
Collapse All
Properties
Variables
Arguments
Imports
50%
Output
Search
Execution started for file: Main012
Desktop execution started
11/19/2024 00:00:00 - 11/21/2024 01:27:03
11/19/2024 00:00:00 - 11/21/2024 01:27:06
Activities
Project
Snippets
Error List
Breakpoints & Bookmarks
Orchestrator Not Connected
Cloud Compatible
VB, Windows
ENG
IN
01:51 AM
21-11-2024


HOME
DESIGN
DEBUG
Desktop - UiPath Studio
New
Save
Export As
Debug File
Cut
Copy
Paste
Undo
Redo
Manage Packages
Manage Entities
Test Manager
App/Web Recorder
Computer Vision
User Events
Table Extraction
UI Explorer
Remove Unused
Analyze File
Export to Excel
Publish
Project
Search project (Ctrl+Alt+P)
Desktop
Dependencies (Windows)
UiPath.Excel.Activities = 2.24.3
UiPath.Mail.Activities = 1.23.11
UiPath.System.Activities = 24.10.6
UiPath.Testing.Activities = 24.10.3
UiPath.UIAutomation.Activities = 24.10.6
Entities
Templates
Main012.xaml
project.json
Main012
Main Sequence
Expand All
Collapse All
Properties
Variables
Arguments
Imports
50%
Output
Search
Execution started for file: Main012
Desktop execution started
11/19/2024 00:00:00 - 11/21/2024 01:27:03
11/19/2024 00:00:00 - 11/21/2024 01:27:06
Activities
Project
Snippets
Error List
Breakpoints & Bookmarks
Orchestrator Not Connected
Cloud Compatible
VB, Windows
ENG
IN
01:51 AM
21-11-2024


HOME
DESIGN
DEBUG
Desktop - UiPath Studio
New
Save
Export As
Debug File
Cut
Copy
Paste
Undo
Redo
Manage Packages
Manage Entities
Test Manager
App/Web Recorder
Computer Vision
User Events
Table Extraction
UI Explorer
Remove Unused
Analyze File
Export to Excel
Publish
Project
Search project (Ctrl+Alt+P)
Desktop
Dependencies (Windows)
UiPath.Excel.Activities = 2.24.3
UiPath.Mail.Activities = 1.23.11
UiPath.System.Activities = 24.10.6
UiPath.Testing.Activities = 24.10.3
UiPath.UIAutomation.Activities = 24.10.6
Entities
Templates
Main012.xaml
project.json
Main012
Main Sequence
Expand All
Collapse All
Variables
Arguments
Imports
50%
Properties
Output
Search
Execution started for file: Main012
Desktop execution started
11/19/2024 00:00:00 - 11/21/2024 01:27:03
11/19/2024 00:00:00 - 11/21/2024 01:27:06
Activities
Project
Snippets
Error List
Breakpoints & Bookmarks
Orchestrator Not Connected
Cloud Compatible
VB, Windows
ENG
IN
01:52 AM
21-11-2024

REFERENCE

1. UiPath Documentation - Email Automation

- Learn how to automate emails, including notification setups for contract expiry.
- **Link**: Email Automation with UiPath

2. UiPath Academy - Beginner to Advanced Courses

- Free online courses to get you up to speed on creating automation workflows, including contract expiry notifications.

Link: https://www.uipath.com/

3. UiPath Community Forum - Discussion and Examples

- Explore community discussions and examples where RPA professionals share their contract notification bot projects and solutions.
- **Link**: UiPath Forum

4. UiPath Marketplace - Templates & Components

- Access pre-built templates and workflows from the UiPath Marketplace that can be used for contract expiration notification.
- **Link**: UiPath Marketplace

5. Automation Anywhere - Documentation & Knowledge Base

- Find guides and articles on building automation workflows for contract expiry notifications, as well as integrating with email systems.
- **Link**: Automation Anywhere Docs

6. GitHub Repositories for RPA Projects

- Explore open-source repositories that contain code or workflows for automating contract expiry notifications.
- **Link**: RPA on GitHub

7. Medium Blog - RPA for Contract Management

- A collection of articles discussing contract management automation, including contract expiry notifications, on Medium.
- **Link**: Medium RPA Articles

8. YouTube Tutorials on RPA Automation

- Watch tutorials and step-by-step guides on building workflows, including contract expiry notification bots, using UiPath or other RPA tools.
- **Link**: UiPath YouTube Tutorials

9. UiPath Studio - Data Manipulation and Date Functions

- Learn how to manipulate data in Excel or databases and perform date comparisons to identify contracts nearing expiration.
- **Link**: UiPath Data Manipulation Activities

10. Automation in Contract Lifecycle Management - Case Studies

- Explore case studies on using RPA for contract lifecycle management and expiration alerts, which can provide insights into similar automation