# TAXA BOT

## A PROJECT REPORT

*Submitted by*

**P JAGANAATH (220701095)**

*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 COLLEGECHENNAI - 602105**

## BONAFIDE CERTIFICATE

Certified that this project report **"TAXA BOT"** is the Bonafede work of **"P JAGANAATH (220701095)"** who carried out the project work for the subject OAI1903-Introduction to Robotic Process Automation under my supervision.


Mrs. J. Jinu Sophia

**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_____.

## TABLE OF CONTENTS

## 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 guides**, Mrs. J. Jinu Sophia, M.E., (Ph.D)** Assistant Professor (SG), 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. 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.

**P JAGANAATH (220701095)**

# ABSTRACT

This project aims to automate the task of sending messages using Telegram by utilizing UiPath and application interaction with the Telegram Bot API and the desktop application. The solution applies UiPath's automation capability to locate the chat elements dynamically using the phone number or the names of the chats. It uses the UI Explorer in order to validate and customize the selectors to target the chat elements reliably. It integrates Click, Find Children, and Send Message to completely automate workflows. It sends messages accurately by parsing JSON responses and fetching chat details. It is very reliable in real-world implementations because it supports robust error handling and application state validation. It is the best system for businesses and organizations that need to engage customers effectively through Telegram.

# LIST OF TABLES

| Field Name | Data Type | Description |
|---|---|---|
| **Extract Table** | Data table | Used to extract the contents form the table |
| **Message** | string | Output message is displayed in the screen |

**Description:**

It enables the automation to retrieve and process recipient details such as phone numbers, chat IDs, names, and messages from various sources like Excel, databases, or JSON responses. The extracted data is used to personalize messages, track delivery statuses, and manage error handling efficiently. By iterating through the Data Table, the workflow facilitates dynamic message customization, robust logging, and seamless integration with the Telegram Bot API or UI-based automation, ensuring scalable and reliable communication.

## LIST OF FIGURES

**API END POINT:**


**Fig. 1**

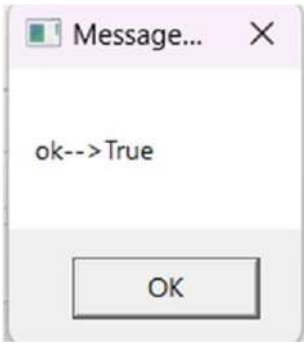



**Fig. 2**

**OUTPUT:**

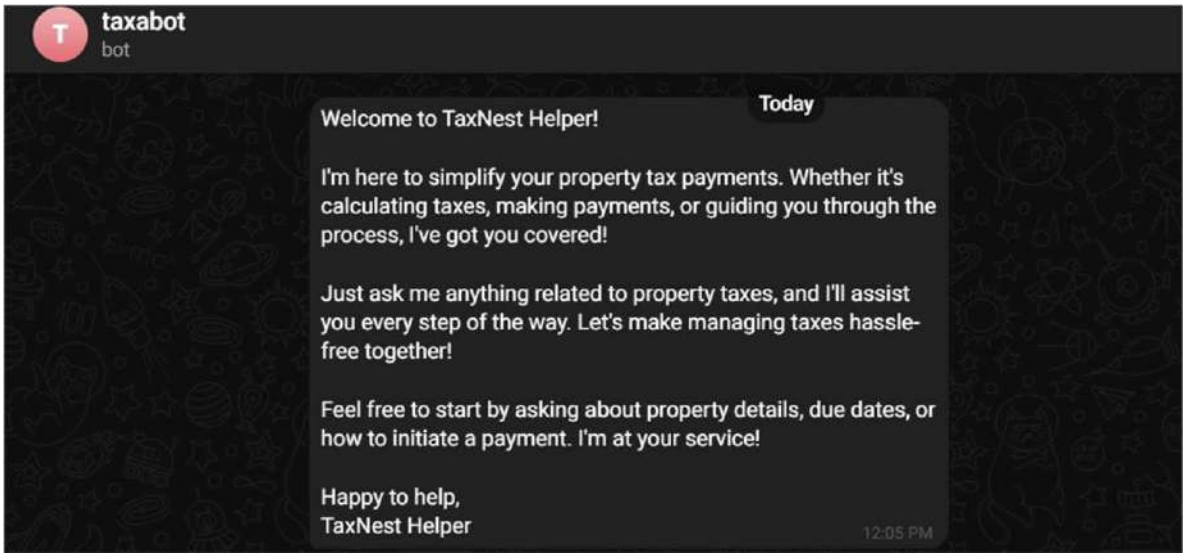



**Fig. 3**

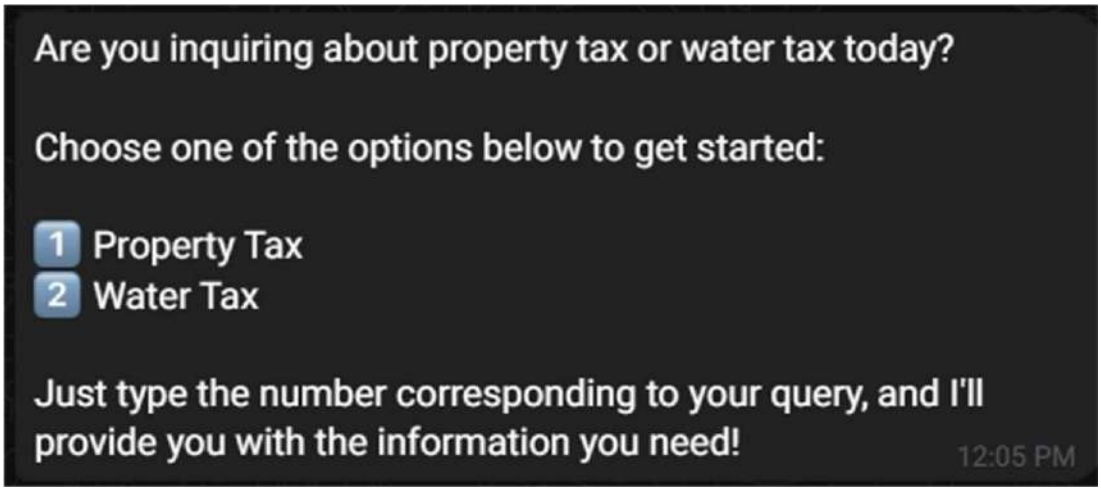

Are you inquiring about property tax or water tax today?
Choose one of the options below to get started:
1 Property Tax
2 Water Tax
Just type the number corresponding to your query, and I'll provide you with the information you need!
12:05 PM

**Fig. 4**

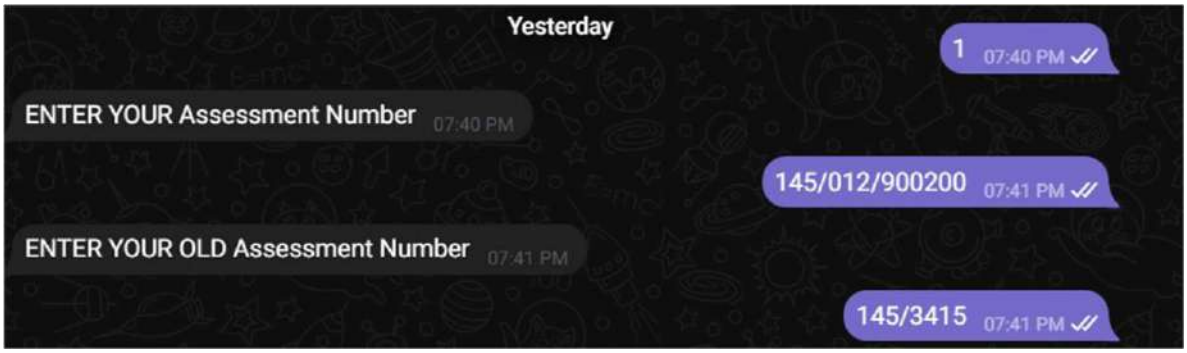

Yesterday
1 07:40 PM
ENTER YOUR Assessment Number 07:40 PM
145/012/900200 07:41 PM
ENTER YOUR OLD Assessment Number 07:41 PM
145/3415 07:41 PM

**Fig. 5**

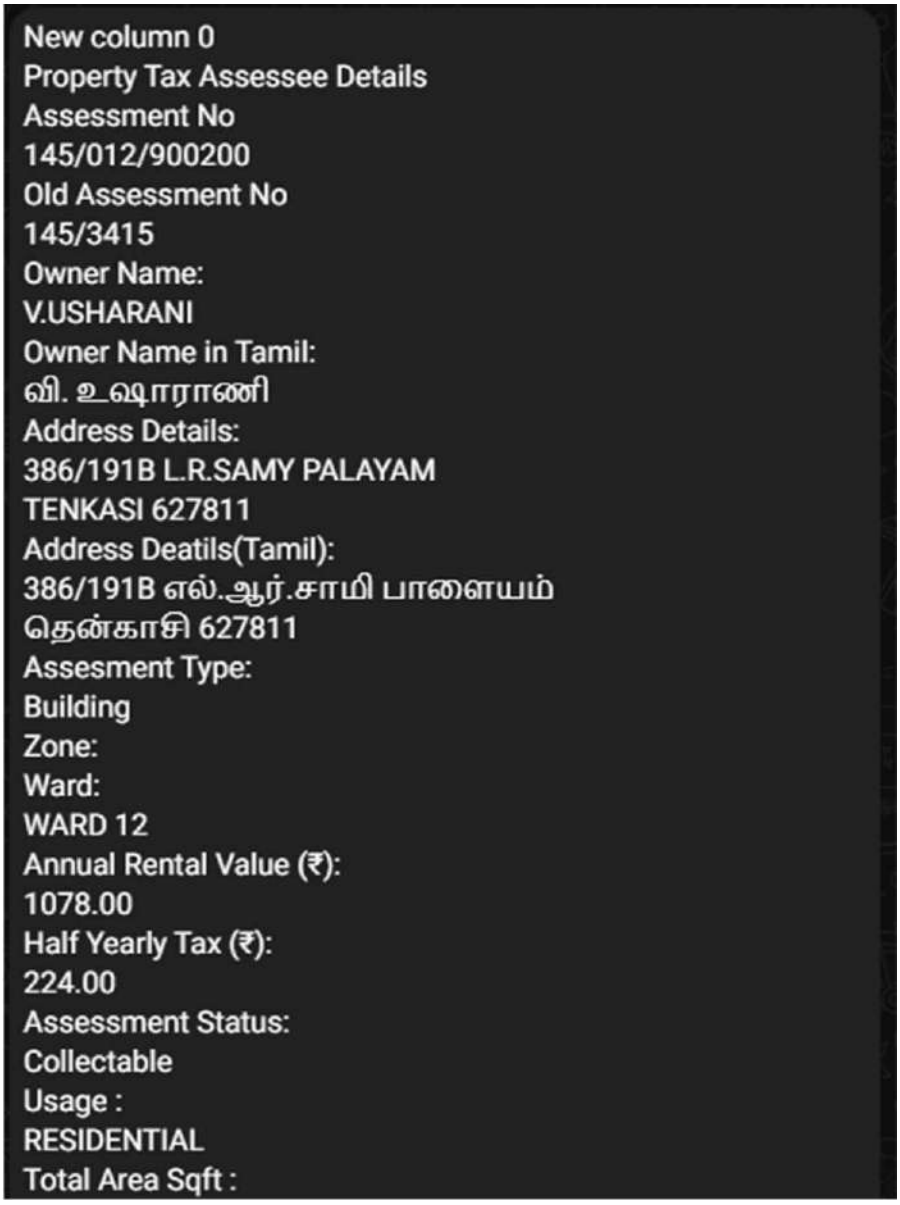

New column 0
Property Tax Assessee Details
Assessment No
145/012/900200
Old Assessment No
145/3415
Owner Name:
V.USHARANI
Owner Name in Tamil:
வி. உஷாராணி
Address Details:
386/191B L.R.SAMY PALAYAM
TENKASI 627811
Address Deatils(Tamil):
386/191B எல்.ஆர்.சாமி பாளையம்
தென்காசி 627811
Assesment Type:
Building
Zone:
Ward:
WARD 12
Annual Rental Value (₹):
1078.00
Half Yearly Tax (₹):
224.00
Assessment Status:
Collectable
Usage :
RESIDENTIAL
Total Area Sqft :

**Fig. 6**

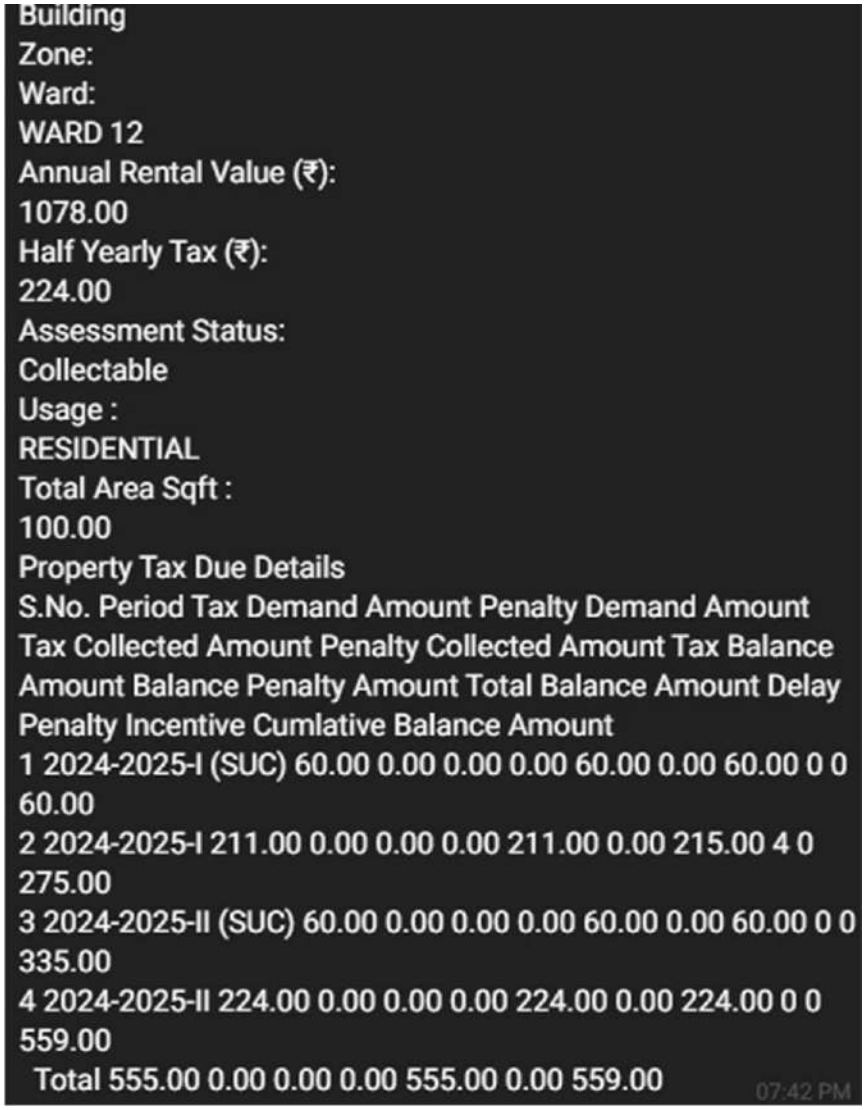

Building
Zone:
Ward:
WARD 12
Annual Rental Value (₹):
1078.00
Half Yearly Tax (₹):
224.00
Assessment Status:
Collectable
Usage :
RESIDENTIAL
Total Area Sqft :
100.00
Property Tax Due Details
S.No. Period Tax Demand Amount Penalty Demand Amount Tax Collected Amount Penalty Collected Amount Tax Balance Amount Balance Penalty Amount Total Balance Amount Delay Penalty Incentive Cumlative Balance Amount
1 2024-2025-I (SUC) 60.00 0.00 0.00 0.00 60.00 0.00 60.00 0 0 60.00
2 2024-2025-I 211.00 0.00 0.00 0.00 211.00 0.00 215.00 4 0 275.00
3 2024-2025-II (SUC) 60.00 0.00 0.00 0.00 60.00 0.00 60.00 0 0 335.00
4 2024-2025-II 224.00 0.00 0.00 0.00 224.00 0.00 224.00 0 0 559.00
Total 555.00 0.00 0.00 0.00 555.00 0.00 559.00
07:42 PM

**Fig. 7**

## LIST OF ABBREVIATIONS

| Variable | Abbreviation |
|---|---|
| **CLI** | Command Line Interface |
| **JSON** | JavaScript Object Notation |
| **UI** | User Interface |
| **HTTP** | Hypertext Transfer Protocol |
| **JSON Path** | JSON Pointer for navigating JSON data |
| **EMI** | Equated Monthly Instalment |
| **BOT** | Robot |
| **API** | Application Programming Interface |
| **CSV** | Comma-Separated Values |
| **IDE** | Integrated Development Environment |

# INTRODUCTION

## 1.1 General:

The introduction section explains the purpose and significance of the project. This tells the motivation behind developing the system and, in short, the problem it is seeking to solve.

In addition, this part points out how the project conforms to current trends in technology and automation.

## 1.2 Objective:

The objective of the project is to design and develop an automated system that must be able to interact efficiently through the Telegram platform.

Automating tasks, which include operations such as retrieval of user data, processing requests, and handling messages, can be managed using Telegram APIs and automation tools.

## 1.3 Existing System:

1. The existing system relies on manual interaction with Telegram, which is time-consuming and prone to errors.

2. Users have to interact independently with the features of Telegram without a centralized tool for automation or integration with other systems.

3. There is no scope for scalable and efficient interaction with data through automated workflows.

## 1.4 Proposed System:

1. The proposed system introduces automation by integrating the Telegram platform into a highly advanced automation technique through UiPath.

2. It will provide features of automated message handling, data processing, and tracking interactions using APIs.

3. The proposed system is user-friendly, scalable, and extremely efficient, dealing with the shortcomings of the present system while ensuring faster execution and improved accuracy in handling the tasks.

# LITERATURE REVIEW

## 2.1 General

This chapter discusses the work related to the change in technology influencing the design of the proposed system. Based on its advantages and disadvantages, it gives a clearer picture of how to improvise upon them.

## 2.2 Review of Related Work

*Telegram Bot Integration:*

Existing work on the implementation of Telegram bots for customer support service, collection of data, or notification type of tasks.
API Integration troubles like high traffic, and ensuring real-time responses.
Security problems in API-based systems

*Automation Using UiPath:*

- Case studies on how the product UiPath can be used for automating tasks which can be classified as redundant
- Comparison between UiPath and other RPA tools like Blue Prism and Automation Anywhere
- UiPath activities for process improvement - UI Automation, Web Automation, API Calls, etc.

*JSON and Data Parsing*:

- Resources using JSON as a lightweight data interchange format.
- Tools and techniques for effective parsing and mapping of JSON responses while working with automation workflows
- Limitations of JSON handling and how to overcome in RPA projects.

*UI Interaction Techniques*:

- Research on enhancing UI element recognition in automated tools.
- Impacts of dynamic UI changes with respect to precision.
- Best Practices for Effective Element Targeting and Error Handling.

### 2.3 Limitations of the Current System

- Inadequate API integration between Telegram and automation processes.
- Irregular error handling during messaging due to dynamic interfaces.
- Scaling the bot wasn't particularly scalable to support complex workflows with structured data sets.

### 2.4 Research Gap

- System bridging the capabilities of the Telegram API and robust automation platforms like UiPath
- Scopes for the improvement of the error handling mechanism, including data-driven automation

Fewer examples on real-world situations of end-to-end automation through Telegram bots.

## SYSTEM DESIGN

This subsection introduces the key design elements of the system. It explains how various components of **Taxa Bot** interact, ensuring seamless operation from user input to data retrieval and output.

### 3.1.1 SYSTEM FLOW DIAGRAM

**Description:**

A system flow diagram visually represents the entire flow of **Taxa Bot**, from user interaction to data processing and response generation. It includes:

1. **Input Stage:** The user submits an assessment number via Telegram.
2. **Processing Stage:** The bot communicates with the tax department's website (using APIs or web scraping) to fetch the requested details.
3. **Output Stage:** The bot formats the retrieved data and sends it back to the user as a Telegram message.

### 3.1.1 SYSTEM FLOW DIAGRAM


**Fig .8**

### 3.1.2 ARCHITECTURE DIAGRAM

**Description:**

The architecture diagram illustrates the structural design of **Taxa Bot**. Key components include:

1. **Frontend (Telegram):** Handles user input and displays output.
2. **Backend (Server):** Manages logic, API requests, and web scraping tasks.
3. **Tax Department Integration:** Connects to the official database to retrieve information.
4. **Database (Optional):** Logs user queries and responses for tracking and analytics.
5. **Automation Tools:** UiPath or similar tools to extract data dynamically.
6. This diagram showcases the interaction between these components to achieve efficient tax information retrieval.

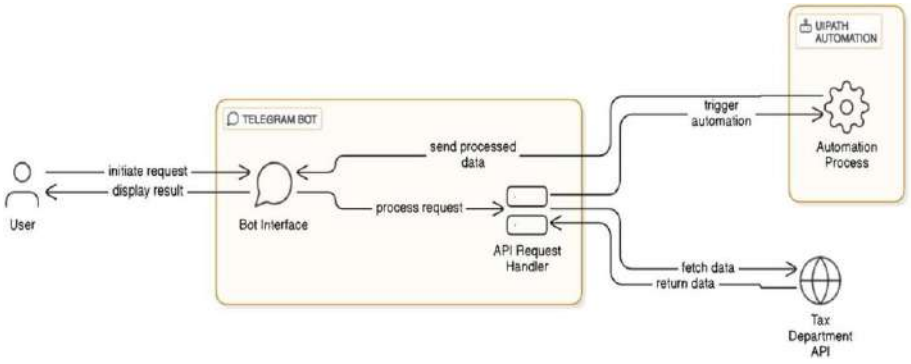



**Fig.9**

This diagram showcases the interaction between these components to achieve efficient tax information retrieval.

## SEQUENCE DIAGRAM

**Description:**

The sequence diagram maps the chronological flow of events and interactions between system components. For **Taxa Bot**, the sequence might look like this:

1. **User Interaction:** The user sends a message with an assessment number.
2. **Telegram Bot:** Forwards the input to the backend server.
3. **Backend:** Processes the input, interacts with the tax department's site, and retrieves data.
4. **Response:** The backend formats the data and sends it back to the Telegram bot.
5. **Output to User:** The bot displays the tax details in the Telegram chat.

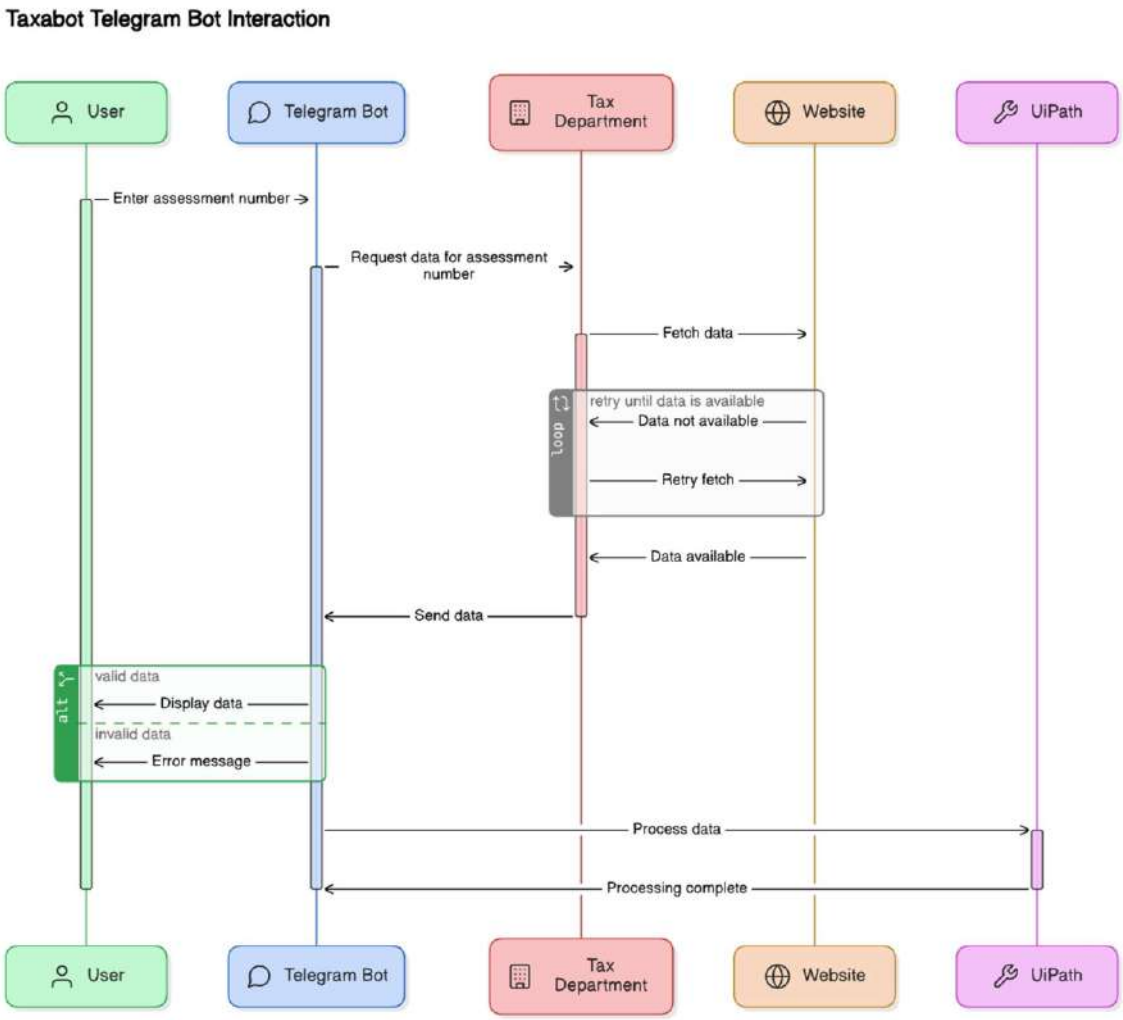
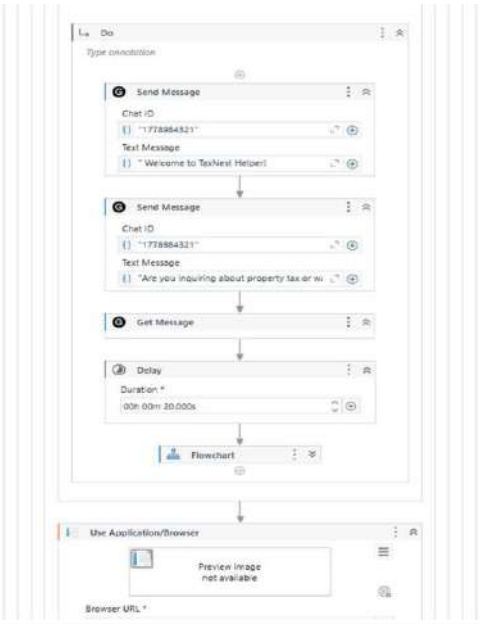



**Fig.10**

This diagram ensures clarity in understanding the order of operations and system behavior under various scenarios

**WORKFLOW**

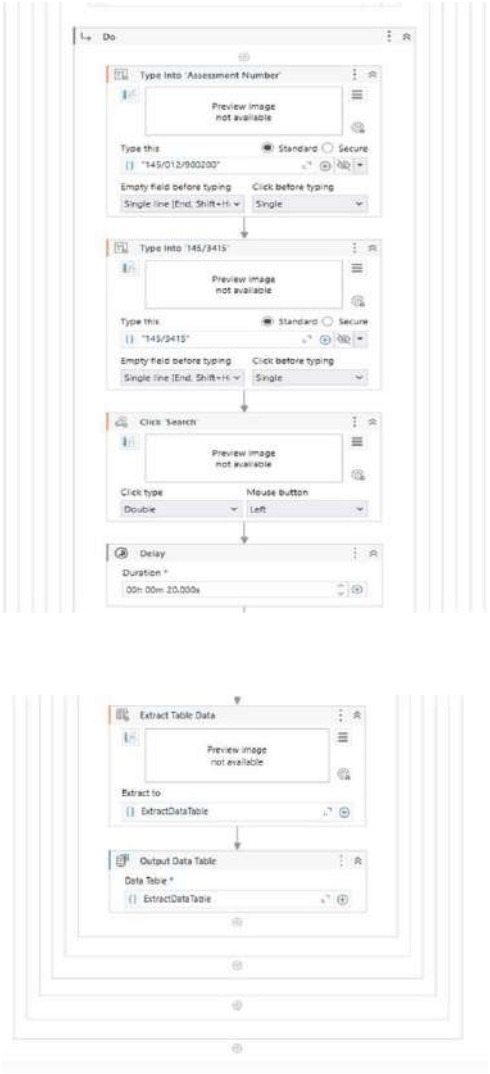

Do
Type into 'Assessment Number'
Preview image not available
Type this
Standard
Secure
"145/012/900200"
Empty field before typing
Click before typing
Single line (End, Shift+Hi
Single
Type into '145/3415'
Preview image not available
Type this
Standard
Secure
"145/3415"
Empty field before typing
Click before typing
Single line (End, Shift+Hi
Single
Click 'Search'
Preview image not available
Click type
Mouse button
Double
Left
Delay
Duration *
00h 00m 20.000s
Extract Table Data
Preview image not available
Extract to
ExtractDataTable
Output Data Table
Data Table *
ExtractDataTable

**Fig.11**

# PROJECT DESCRIPTION

This section provides a comprehensive overview of the approach, methodology, and modules used to develop the automated messaging system for Telegram using UiPath. It explains the system's components, the workflows implemented, and the tools leveraged to achieve the project’s objectives.

## 4.1 METHODOLOGY

The methodology section outlines the steps taken in the development of the automated messaging system, from initial planning to final implementation. The key methodologies adopted in this project include **Robotic Process Automation (RPA)** using **UiPath**, **API Integration** with **Telegram Bot API**, and **UI Automation** for dynamic interaction with desktop applications.

**1. Robotic Process Automation (RPA) with UiPath**

RPA was used to automate the manual process of sending messages on Telegram. UiPath’s capabilities, such as workflow automation, dynamic selector identification, and desktop automation, were leveraged to create an efficient and reliable system.

**2. API Integration with Telegram Bot API**

The integration of the Telegram Bot API with UiPath enabled seamless interaction with the Telegram platform. By using the API, the automation system could send messages, retrieve chat details, and validate interactions without manual intervention.

**3. UI Automation**

UiPath's UI Explorer and dynamic selector functionalities were crucial for automating the process of interacting with the Telegram desktop application. The automation system dynamically identified chat elements based on user input, such as phone numbers or chat names, and used this information to send messages.

**4. Error Handling and Validation**

The methodology included incorporating robust error handling to ensure smooth operations. Error detection mechanisms were implemented to address common issues like incorrect chat selection, connection failures, or failed message delivery. Application state validation was also employed to confirm that the Telegram desktop app was in the correct state before executing any actions.

### 4.1.1 MODULES

The automated messaging system was divided into several modules to manage different aspects of the process. Each module was designed to handle a specific part of the overall workflow, ensuring that the system is both scalable and maintainable. Below are the key modules of the project:

**1. Message Sending Module**

This module handles the core functionality of sending messages to users or groups. It interacts with the Telegram Bot API to send messages based on dynamic inputs (e.g., phone numbers or chat names). The module ensures that messages are sent according to predefined templates or user-specified content.

**2. Chat Identification Module**

This module is responsible for identifying and selecting the correct chat element within the Telegram application. By using UiPath's Find Children activity and custom selectors, it ensures that the correct chat or group is targeted before sending a message.

**3. Data Parsing and Extraction Module**

This module is used to parse incoming data (such as JSON responses from the Telegram Bot API or other external systems) and extract relevant details like chat IDs, names, and other necessary information. It allows the system to dynamically process different chat scenarios.

**4. Monitoring and Reporting Module**

This module provides real-time monitoring of the automation workflow, tracking the status of messages sent, any errors encountered, and the overall performance of the system. It generates reports for analysis and debugging purposes.

# CONCLUSIONS

## 5.1 GENERAL

This project successfully developed an automated messaging system for Telegram using UiPath and the Telegram Bot API. The system effectively automates the process of sending messages, improving efficiency and scalability for businesses.

**Key findings include:**

- *Increased Efficiency*: The automation reduced manual effort, allowing for faster message delivery and improved customer engagement.
- *Seamless Integration*: The integration with the Telegram Bot API enabled dynamic and flexible messaging, with messages sent based on phone numbers or chat names.
- *Error Handling*: Robust error detection and retry mechanisms ensured reliability in real-world use cases.
- *Scalability*: The system is adaptable and can be scaled for various business needs.

However, challenges such as UI automation complexity and API limitations were encountered. Future improvements could include enhanced personalization, cross-platform support, and performance optimizations for handling high volumes of messages.

In conclusion, the system meets its objectives, offering a reliable solution for automated customer engagement on Telegram, with room for further enhancements**.**

# REFERENCES

**Title:** *"The AI Enabled Chatbot Framework for Intelligent Citizen-Government Interaction for Delivery of Services"*

**Authors:** Not explicitly mentioned in the abstract; authored under IEEE Conference Publications.

**Key Focus:** Discusses an AI-enabled chatbot framework for improving citizen-government interactions by delivering efficient public services. It emphasizes automation and natural language processing for seamless service delivery.

**Link:** View Paper

**Advantages:**

- Enhances accessibility of government services.
- Reduces service delivery time.

**Disadvantages:**

- Requires substantial initial development and integration efforts.
- Limited adaptability for complex services.

**Title:** *"The Role of AI Chatbots in Mental Health Related Public Services in a (Post)Pandemic World: A Review and Future Research Agenda"*

**Authors:** Unspecified; featured in IEEE Xplore.

**Key Focus:** Explores the application of AI chatbots in mental health services, particularly in public sectors, analyzing their effectiveness and potential for further advancements.

**Link:** View Paper

**Advantages:**

- Addresses critical gaps in mental health service accessibility.
- Provides a scalable solution for widespread public health challenges.

**Disadvantages:**

- May not effectively handle nuanced or sensitive user inputs.
- Data privacy remains a significant concern.

# APPENDICES

**MAIN Workflow Script (e.g., main.py or Main.xaml):**

- Handles the primary logic of the bot, including message processing, validation of assessment numbers, and communication with external services like the Tamil Nadu tax department's website.

**INPUT Folder:**

- Contains files that serve as input for the automation processes. Examples include:
    - **AssessmentNumbers.xlsx:** A file storing test data or frequently queried assessment numbers for batch processing.

**OUTPUT Folder:**

- Stores logs, results, or fetched tax details for record-keeping and debugging. Examples:
    - **FetchedData.json:** A JSON file with the retrieved tax data.
    - **ExecutionLogs.txt:** Log files capturing the execution flow and errors.