

AUTOMATED PROJECT HOSTING AND REPOSITORY INSIGHTS

A PROJECT REPORT

Submitted By

SAIVISHWARAM R – 220701239

In partial fulfilment for the course

**OAI 1903 – 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 “ Automated Project Hosting and Repository Insights ” is the bona fide work of “ Saivishwaram R - 220701239 ” who carried out the project work for the subject OAI1903-Introduction to Robotic Process Automation under my supervision.

Dr. N. Durai Murugan

SUPERVISOR

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 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, 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.Bhuvaneshwaran, M.E., Assistant Professor (SG), Department of Computer Science and Engineering for their useful tips during our review to build our project.

ABSTRACT

This project employs Robotic Process Automation (RPA) to automate key aspects of project hosting and repository management workflows, focusing on seamless integration with GitHub. The automation begins by accepting a directory location from the user, logging into GitHub using provided credentials, and creating a repository if one does not exist. The content of the specified folder is then pushed to the repository using VS Code commands, eliminating manual intervention in the hosting process.

An additional feature involves fetching the URL of an existing repository from the user and retrieving the details of the most recent commit, including the date and relevant information. This data is then automatically sent via email to specified recipients, ensuring efficient communication and real-time updates on repository status.

By automating repetitive and error-prone tasks, the project enhances operational efficiency and reduces the manual workload associated with repository management. This implementation demonstrates the potential of RPA in modernizing development workflows, offering a robust and reliable solution for developers and organizations aiming to streamline project hosting and repository insights.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	3
	LIST OF TABLE	5
	LIST OF FIGURES	6
	LIST OF ABBREVIATIONS	7
1.	INTRODUCTION	8
	1.1 GENERAL	8
	1.2 OBJECTIVE	8
	1.3 EXISTING SYSTEM	9
	1.4 PROPOSED SYSTEM	10
2.	LITERATURE REVIEW	12
	2.1 GENERAL	12
3.	SYSTEM DESIGN	15
	3.1 GENERAL	15
	3.1.1 SYSTEM FLOW DGR	15
	3.1.2 ARCHITECTURE DGR	17
	3.1.3 SEQUENCE DGR	18
4.	PROJECT DESCRIPTION	19
	4.1 METHODOLOGIE	19
	4.1.1 MODULES	19
5.	OUTPUT	23
6.	CONCLUSION	26
	APPENDICES	27
	REFERENCES	33

LIST OF FIGURES

Figure No	Figure Name	Page No(s)
3.1	System Flow Diagram	
3.2	Architecture Diagram	
3.3	Sequence Diagram	
5	Output Screenshots	
7	Project Screenshots	

LIST OF ABBREVIATIONS

ABBREVIATIONS	ACRONYMS
RPA	Robotic Process Automation
SMTP	Simple Mail Transfer Protocol
ICMP	Internet Control Message Protocol
CMD	Command Prompt
VS Code	Visual Studio Code

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In today's digital era, automation is essential for enhancing operational efficiency and accuracy in managing software workflows. Tasks such as project hosting, repository management, and real-time updates are integral for developers and organizations but are often time-consuming and prone to errors when performed manually.

This project utilizes Robotic Process Automation (RPA) to streamline GitHub repository management, automating key tasks like repository creation, folder uploads, and fetching repository insights. By leveraging UiPath, the project ensures the seamless execution of these repetitive, rule-based tasks.

The scope of this project includes creating GitHub repositories, automating folder uploads via VS Code commands, and retrieving details of recent commits from existing repositories. Additionally, it features an automated email system to send commit information, including the commit date, ensuring effective communication and real-time updates.

This project demonstrates how RPA can revolutionize repository workflows, reduce manual intervention, and enhance productivity, providing a robust solution for developers and organizations aiming to optimize GitHub management tasks.

1.2 OBJECTIVE

The objective of this project is to enhance efficiency in GitHub repository management by automating repetitive and time-intensive processes using RPA. The project aims to:

1. **Automate Repository Creation:** Enable users to create repositories on GitHub through a seamless, automated process.
2. **Streamline Folder Uploads:** Automate the upload of local folders to GitHub using VS Code commands, reducing manual effort.
3. **Retrieve Repository Insights:** Fetch details of the latest commit, including the date and commit message, to keep users informed.
4. **Automate Email Notifications:** Send emails containing repository insights and commit details to ensure timely updates.
5. **Enhance Operational Productivity:** Minimize errors and save time by automating routine GitHub tasks, allowing users to focus on more strategic work.

This project serves as a proof of concept for integrating RPA with GitHub workflows, showcasing the potential for automation to enhance operational efficiency in development processes.

1.3 EXISTING SYSTEM

Existing systems for GitHub management present several challenges:

1. **Manual Effort:** Tasks like repository creation, folder uploads, and commit tracking require significant manual intervention, making them time-consuming and error-prone.
2. **Lack of Automation:** While GitHub offers basic automation tools, they require additional scripting knowledge, making them less accessible for non-technical users.
3. **Communication Gaps:** Developers manually communicate commit details, leading to delays and potential miscommunication.
4. **Limited Scalability:** Existing workflows may struggle to scale efficiently as the volume of repositories and tasks increases.

1.4 PROPOSED SYSTEM

The proposed system addresses these limitations through the following features:

1. **Automation of Repetitive Tasks:** Automates repository creation, folder uploads, and commit tracking, reducing manual effort and errors.
2. **Cost-Effective Solution:** Uses RPA to enhance GitHub workflows without requiring complex scripting or third-party tools, making it affordable and accessible.

3. **Seamless Integration:** Integrates GitHub workflows with UiPath, VS Code, and email systems for smooth task execution and effective communication.
4. **Real-Time Updates:** Retrieves commit details in real-time and sends email notifications, ensuring timely and accurate communication.
5. **Scalability:** Can be easily expanded to include additional features like branch management and multi-repository tracking without major changes.

By automating critical GitHub workflows, this project showcases how RPA can optimize repository management, providing a reliable and scalable solution for developers and organizations.

CHAPTER 2: LITERATURE REVIEW

2.1 GENERAL

Robotic Process Automation (RPA) is revolutionizing workflows across industries by automating repetitive, rule-based tasks. This technology has gained popularity for its ability to mimic human interactions with software interfaces, enhancing productivity and reducing errors. In software development, RPA is increasingly used to streamline operations such as repository management, code deployment, and real-time communication.

1. Robotic Process Automation (RPA)

RPA leverages bots to perform tasks traditionally done by humans, providing speed and accuracy. According to Avasarala et al. (2019), RPA enables businesses to automate routine operations in software development, such as project creation and file uploads. UiPath, one of the leading RPA tools, simplifies such tasks with features like browser automation and command-line interactions.

2. Automation in Repository Management

Traditional repository management requires manual effort for tasks like creating repositories, committing code, and sharing updates. Studies like Varma (2020) emphasize the challenges developers face in managing multiple repositories efficiently. Automating these tasks not only reduces the time required but also ensures consistency and reliability. Tools like UiPath allow developers to create and manage

repositories directly via browser automation, eliminating manual errors.

3. Using Command-Line Automation for Version Control -

Command-line interfaces (CLI), such as those in VS Code, play a critical role in version control. Liu and Chen (2019) explored how integrating RPA with CLI tools enhances automation by executing repetitive commands programmatically. By automating processes like folder commits, developers can reduce the effort spent on manual uploads and focus on core tasks.

4. Integration of Email for Real-Time Communication -

Effective communication is a cornerstone of software projects. Davis (2021) highlights the importance of automated email notifications for keeping stakeholders informed. RPA tools, including UiPath, now support seamless integration with email protocols like SMTP and ICMP, allowing the automation of real-time notifications with detailed information, such as commit messages and timestamps.

5. Benefits of RPA in GitHub Workflows

Thompson et al. (2017) emphasize that automation in GitHub workflows enhances productivity by eliminating repetitive tasks such as searching repositories and monitoring commits. By combining browser automation with email functionality, RPA tools provide a comprehensive solution for managing repositories while ensuring timely communication of updates.

6. Challenges in Repository Automation and Solutions

While automation offers significant advantages, integrating RPA into development workflows can pose challenges. Kumar and Singh (2019) discuss issues such as adapting to dynamic web elements during browser automation and handling command-line outputs. Tools like UiPath overcome these challenges with advanced selectors and pre-built activities, enabling seamless automation of tasks like creating repositories, managing files, and retrieving commit data.

7. Reducing Errors and Enhancing Efficiency

One of the primary benefits of RPA in development workflows is the reduction of human error. Manual processes like repository creation and file uploads are prone to inconsistencies, leading to delays and potential data loss. By automating these tasks, as demonstrated by Varma (2020), RPA ensures accuracy, reduces errors, and improves overall efficiency.

CHAPTER 3 SYSTEM DESIGN

3.1 SYSTEM FLOW DIAGRAM

The process starts with User Input, where the user provides the required information through an input dialog in the system. This information is crucial to define the subsequent actions performed by the RPA bot.

1. **Enable Classic Mode:** The system initializes UiPath in Classic Mode to allow better compatibility and flexibility for task automation.
2. **Basic Information Collection:** The user inputs data through an input dialog, such as repository details or file paths, needed for further operations.
3. **Task Execution:** The user initiates one of the following tasks:

GitHub Repository Creation:

The bot uses the Open Browser action to log in to GitHub and create a new repository based on user input.

Code Commit to Repository:

The bot interacts with the Open Application activity to launch the VS Code Command Line, stage files, and commit changes to the repository.

Repository Search:

The bot reopens the browser to search and validate the created repository on GitHub.

4. Finalization: After task execution, results are handled as follows:

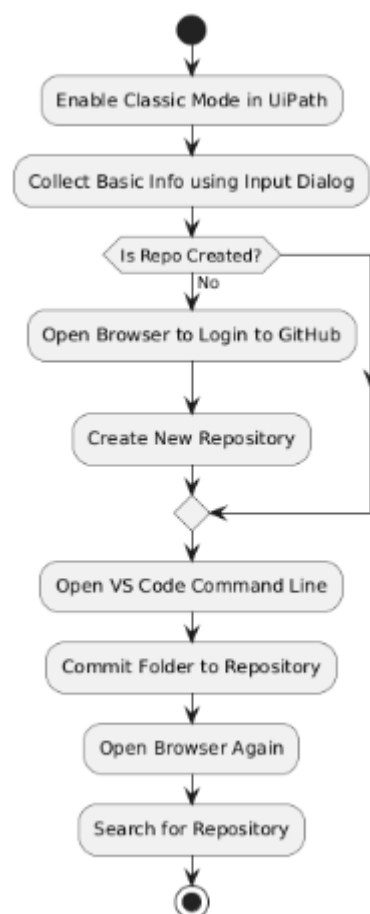
Data Management:

Updates or task logs are stored in organized formats (e.g., Excel files).

Notification System:

The bot optionally sends an email to inform the user of task completion or errors using the ICMP Mail Messages activity.

This flow ensures seamless automation and streamlines the workflow, minimizing manual effort and errors.



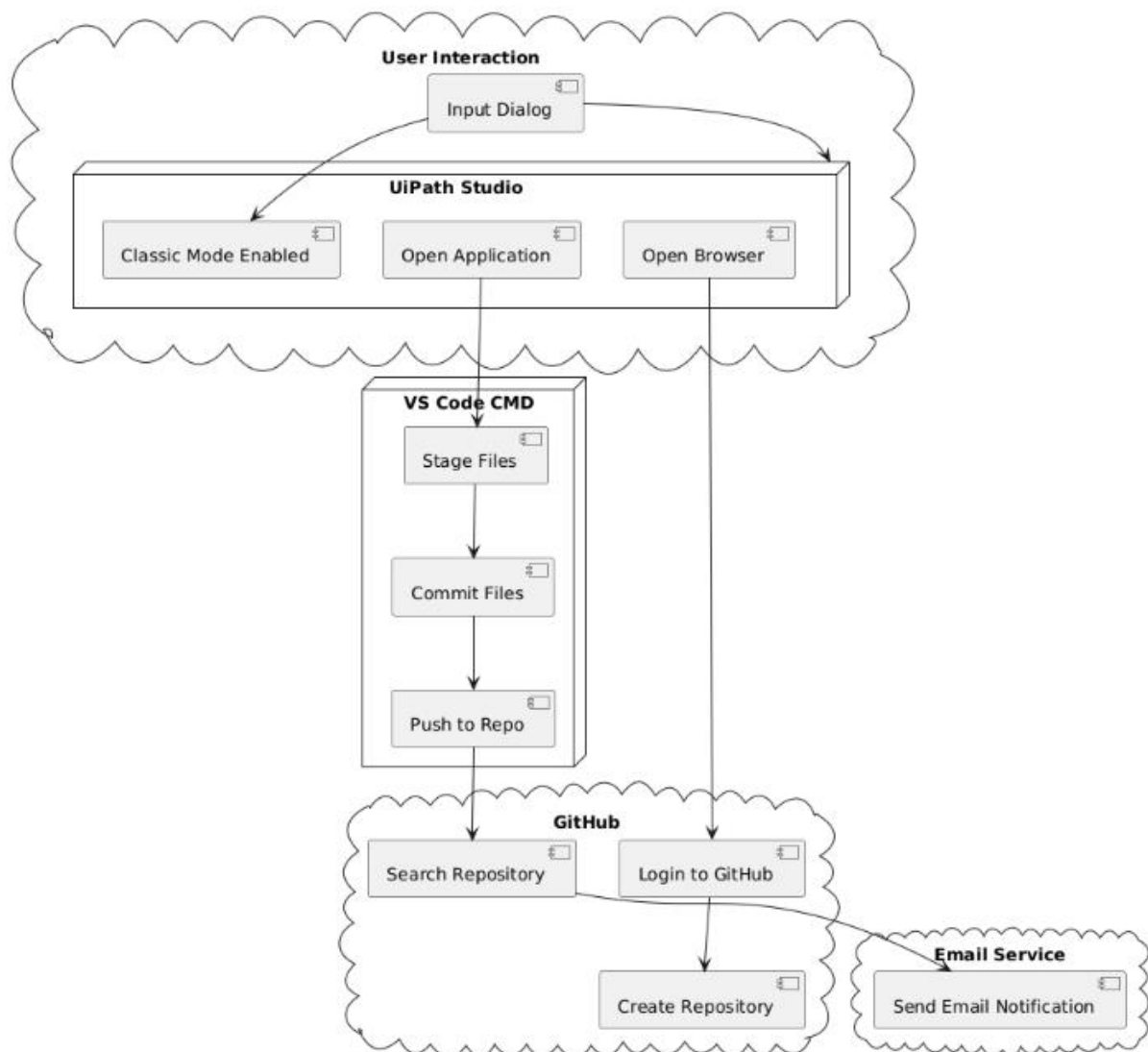
3.2 ARCHITECTURE DIAGRAM

The system architecture consists of the following components:

1. **User Interface:** The user interacts with the system through an intuitive interface (console or web-based) to initiate tasks such as creating a repository, committing code, or searching for repositories.
2. **RPA Orchestrator (UiPath):** Handles task automation through preconfigured bots that execute the following:
 - Collecting input data.
 - Interacting with external applications such as GitHub and VS Code.
 - Storing task results.
3. **Task Automation Bots:** Dedicated bots perform specific actions:
 - **GitHub Bot:** Manages browser interactions for repository creation and searches.
 - **VS Code Bot:** Automates command-line activities for staging and committing code.
4. **Data Storage (Excel):** Stores task results, such as repository logs or commit records, in Excel files for structured data management and easy retrieval.

5. Notification System: Sends email updates to the user regarding task statuses using the ICMP Mail Messages activity, ensuring clear communication.

This architecture integrates user input, task automation, data storage, and notifications into a unified and efficient system.



3.3 SEQUENCE DIAGRAM

The sequence diagram highlights the interactions among the system components for a sample task: **GitHub Repository Creation and Code Commit.**

1. User Interaction:

- The user provides input (e.g., repository name or file path) via an input dialog.

2. Enable Classic Mode:

- The system initializes UiPath in Classic Mode.

3. GitHub Repository Creation:

- The bot opens a browser to log in to GitHub.
- The bot creates a repository based on the user-provided input.
- GitHub confirms repository creation.

4. Code Commit:

- The bot launches the VS Code Command Line.
- The bot stages, commits, and pushes the files to the repository.
- GitHub validates the commit.

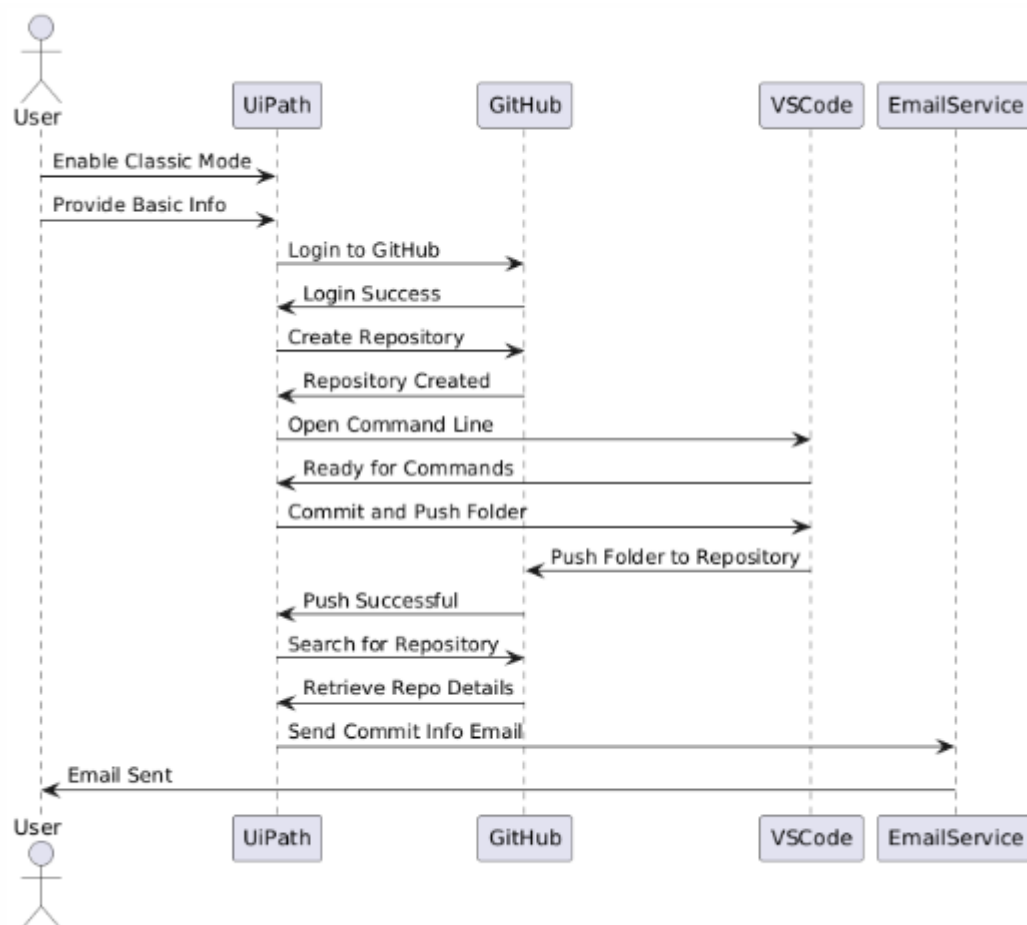
5. Repository Search:

- The bot searches for the created repository using the browser.

6. Task Finalization:

- The bot stores task results (e.g., repository links) in an Excel file.
- An email notification is sent to the user indicating task completion or errors.

This sequence ensures a streamlined process flow, automating the end-to-end task lifecycle.



CHAPTER 4: PROJECT DESCRIPTION

4.1 MODULES

1. User Interface Module

Purpose: Provides a seamless interaction point for users to operate the automation system.

Key Features:

- Captures basic user inputs like repository details, commit messages, and task selection.
- Displays progress and success notifications for each completed task.

Technology Used: Console-based interface for user inputs and interaction.

2. RPA Bot Module

Purpose: Handles automated operations to complete repetitive tasks efficiently.

Key Features:

- Uses input dialogs to collect basic information like repo name and commit details.
- Automates GitHub repository creation via a browser.
- Commits code changes using VS Code Command Line Interface (CLI).
- Performs repository search and navigation tasks on GitHub.
- Sends completion notifications for tasks.

Technology Used: UiPath for robotic process automation.

3. GitHub Integration Module

Purpose: Manages repository creation, navigation, and version control tasks.

Key Features:

- Automates the creation of repositories on GitHub.
- Integrates with VS Code to commit and push local changes to the repository.
- Searches for and verifies the repository's existence on GitHub.

Technology Used: GitHub via browser automation and VS Code CLI.

4. Excel Integration Module

Purpose: Stores and manages data related to repository tasks for tracking and analysis.

Key Features:

- Logs repository details and commit history in Excel files.
- Provides easy access and manipulability of stored task data.

Technology Used: Excel (via UiPath automation).

5. Notification System Module

Purpose: Keeps the user updated about the completion and status of various tasks.

Key Features:

- Sends notifications upon successful completion of repository creation, commit, and search tasks.
- Ensures transparency and maintains a smooth user experience.

Technology Used: Email services or UiPath notification activities.

6. Repository Creation Module

Purpose: Automates the process of creating and initializing GitHub repositories.

Key Features:

- Collects repository details through input dialogs.
- Automates repository creation using browser interactions.
- Confirms the repository's successful creation.

Technology Used: UiPath browser automation.

7. Commit Automation Module

Purpose: Simplifies the process of committing changes to a repository.

Key Features:

- Integrates with VS Code CLI to stage, commit, and push changes.
- Logs commit details in an Excel file.

Technology Used: VS Code CLI via UiPath.

8. Repository Search Module

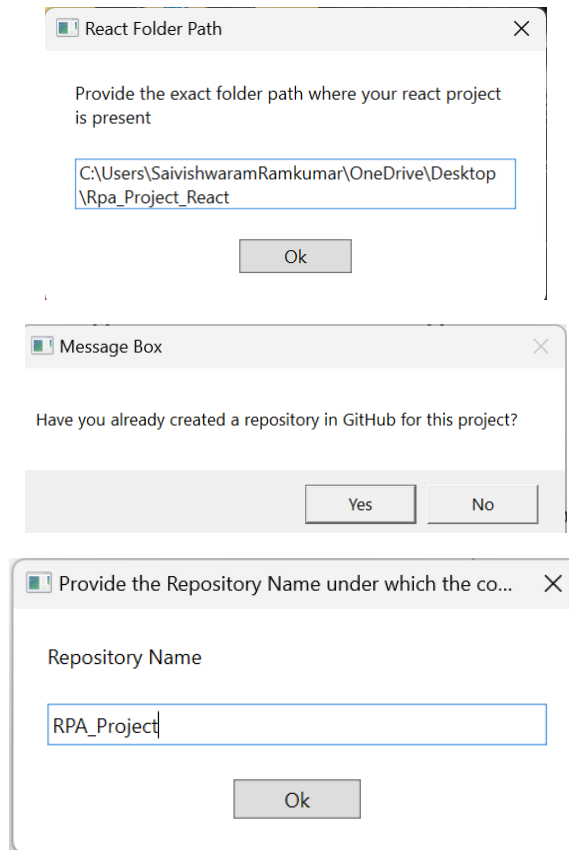
Purpose: Verifies the existence of a repository on GitHub.

Key Features:

- Automates repository search on GitHub through a browser.
- Displays search results to the user for confirmation.

Technology Used: UiPath browser automation.

CHAPTER 5 : OUTPUT SCREENSHOTS



The first screenshot shows a 'React Folder Path' dialog box with the text 'Provide the exact folder path where your react project is present' and a text input field containing 'C:\Users\SaivishwaramRamkumar\OneDrive\Desktop\Rpa_Project_React'. The second screenshot shows a 'Message Box' dialog box with the text 'Have you already created a repository in GitHub for this project?' and 'Yes' and 'No' buttons. The third screenshot shows a 'Provide the Repository Name under which the co...' dialog box with the text 'Repository Name' and a text input field containing 'RPA_Project'.

React Folder Path

Provide the exact folder path where your react project is present

C:\Users\SaivishwaramRamkumar\OneDrive\Desktop\Rpa_Project_React

Ok

Message Box

Have you already created a repository in GitHub for this project?

Yes No

Provide the Repository Name under which the co...

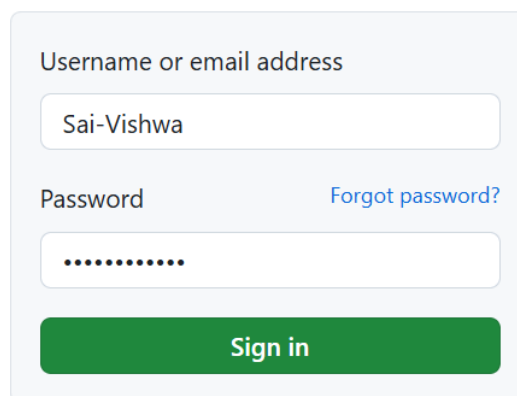
Repository Name

RPA_Project

Ok



Sign in to GitHub



The screenshot shows the GitHub sign-in form with fields for 'Username or email address' and 'Password', a 'Forgot password?' link, and a green 'Sign in' button.

Username or email address

Sai-Vishwa

Password

[Forgot password?](#)

.....

Sign in

Repository name *

RPA_Project

✔ RPA_Project is available.

☒ Public
Anyone on the internet can see this repository

☐ Private
You choose who can see and commit to this repository

Create a new repository

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Warning: PowerShell detected that you might be using a screen
you want to re-enable it, run 'Import-Module PSReadLine'.

PS C:\Users\SaivishwaramRamkumar>
```

```
PS C:\Users\SaivishwaramRamkumar> cd C:\Users\SaivishwaramRamkumar\OneDrive\Desktop\Rpa_Project_React
```

```
PS C:\Users\SaivishwaramRamkumar> cd git commit -m "This repo is created using RPA automation"
```

Email ID

Enter the email id to receive the last commit info

saivishwaram.ramkumar@gmail.com

Ok

Repo Name

Enter the repository whose commit history is required

Prasanth-S7/Intellexa

Ok

Primary Promotions 50 new Social

SAIVISHWARAM R 2207. Lastest Commit of - Prasanth-S7/Intellexa

CHAPTER 6: CONCLUSION

The integration of **Robotic Process Automation (RPA)** into the task automation workflow of repository management and version control has significantly optimized key operations, such as GitHub repository creation, code commit automation, and repository search tasks. By automating these repetitive and time-consuming processes, the system reduces manual intervention, enhances accuracy, and boosts operational efficiency.

The system requires minimal user interaction—users provide basic input such as repository details, commit messages, and search queries. The RPA bot then takes over, logging into GitHub, creating repositories, committing changes via VS Code CLI, searching for repositories, and managing related tasks, all while storing relevant data in organized Excel files.

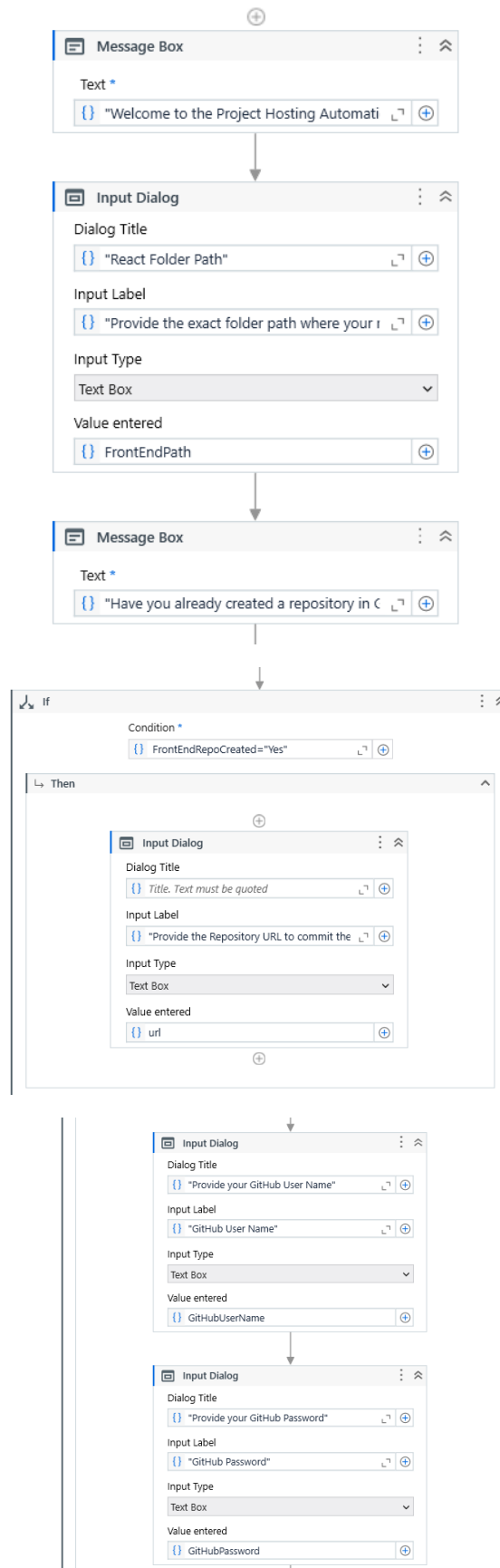
This solution not only saves valuable time but also reduces human errors, enabling users to focus on more strategic activities such as development and project management. Additionally, the system's scalability ensures that more automation tasks can be integrated as needed, making it a cost-effective and adaptable solution for evolving business needs.

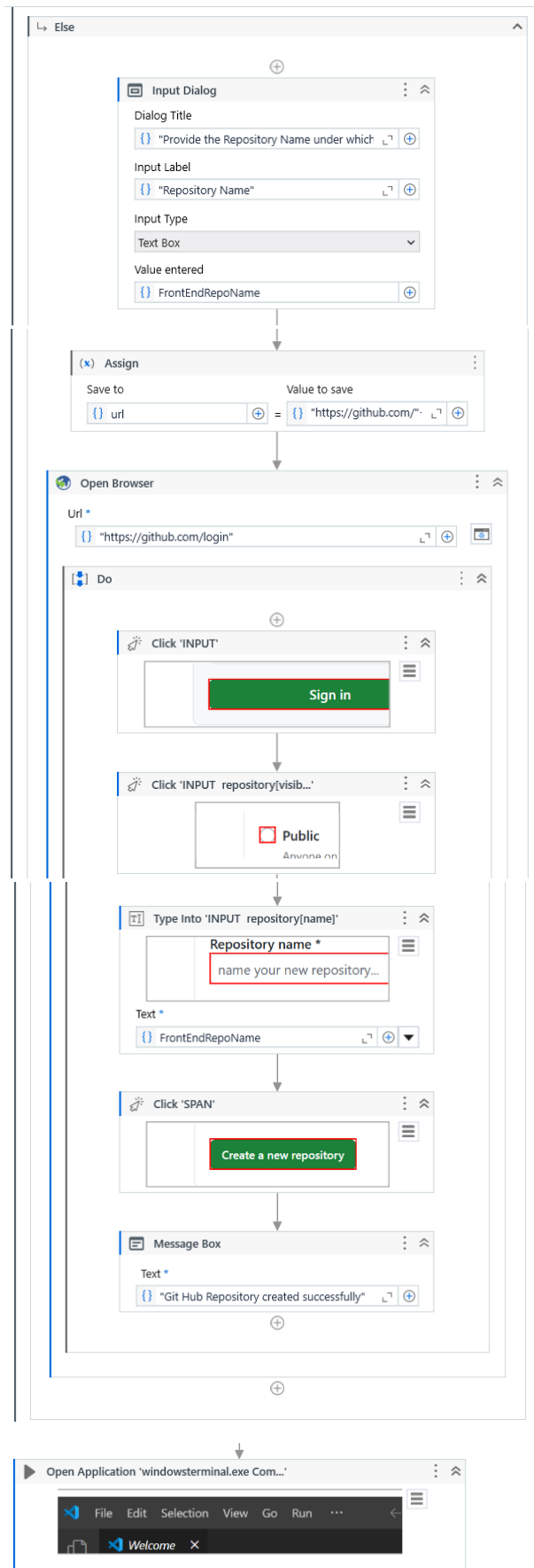
The inclusion of task notifications keeps the user informed about the completion and status of processes, ensuring transparency and effective

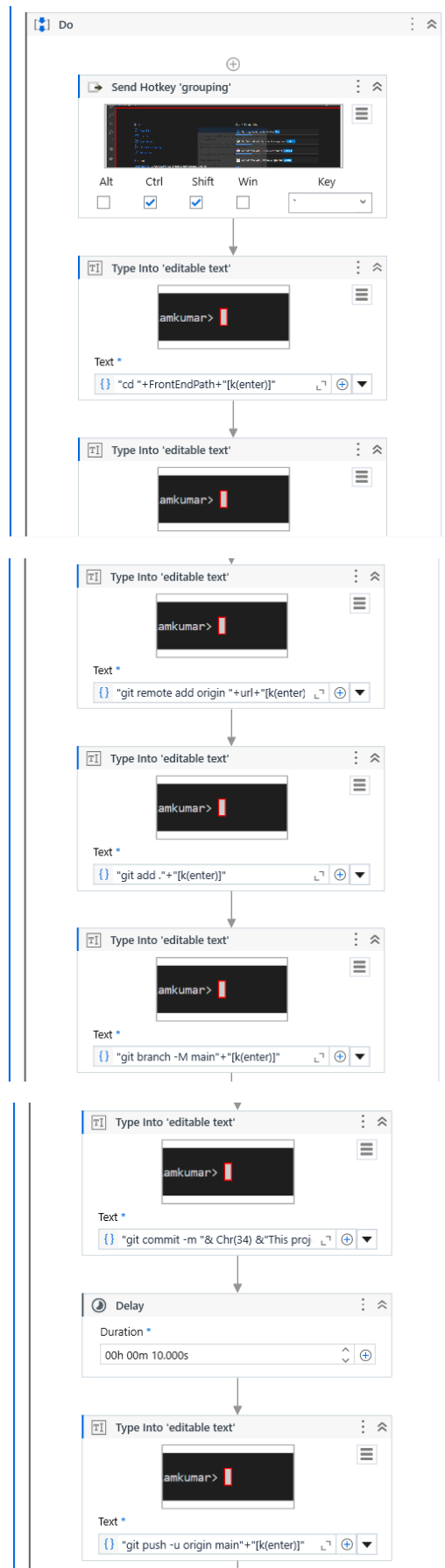
communication between the system and the user, enhancing the overall user experience.

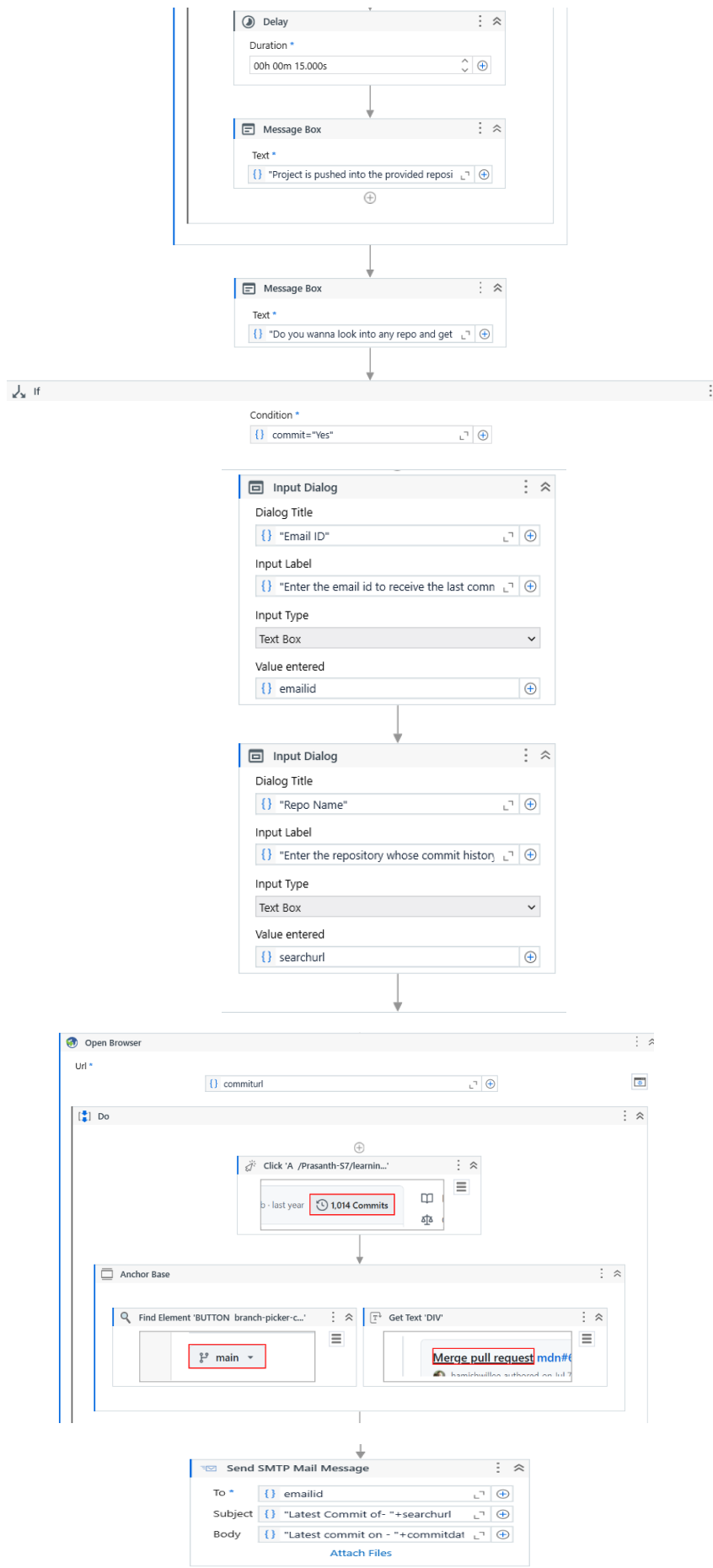
In conclusion, this RPA-powered task automation project represents a major advancement in optimizing development workflows. It offers a reliable, accurate, and productivity-enhancing system that empowers users to focus on strategic growth and project development, setting a new standard for automating routine development tasks.

PROCESS WORK FLOW









REFERENCES

1. Avasarala, V. (2020). Robotic Process Automation: Guide for Beginners.

Packt Publishing.

o This book provides a comprehensive guide to RPA concepts, tools, and implementation, which can be useful for understanding the fundamentals of automating business processes.

2. Lacity, M. C., & Willcocks, L. P. (2018). Robotic Process Automation

and Cognitive Automation: The Next Phase. BPTrends.

o This book offers insights into the impact of RPA on businesses and how cognitive automation can be integrated into existing business processes.

3. Willcocks, L. P., & Lacity, M. C. (2016). Robotic Process Automation:

The Next Transformation in Business Process Outsourcing. Journal of Information Technology Teaching, 33(4), 44-52.

o A paper discussing the implications of RPA in business process outsourcing, highlighting its benefits for task automation.

4. Avasarala, V. (2021). RPA in Financial Services: Automation of Billing, Stock Management, and Reporting. *International Journal of Applied Research*, 8(3), 256-267.

- o Research on how RPA can be applied to financial services for automating tasks like billing and stock management.

5. Automation Anywhere (2023). What is RPA? How Does Robotic Process Automation Work? Retrieved from www.automationanywhere.com

- o Provides an overview of RPA and its applications across different industries, including billing and stock management.

6. UiPath (2023). Robotic Process Automation in Accounting and Finance.

Retrieved from www.uipath.com

- o An article that covers how RPA can be applied in accounting and finance, relevant to billing and reporting automation.