FULL STACK PROJECT SETUP AUTOMATION BOT

A PROJECT REPORT

Submitted by

SAFEEKUR RAHMAN A.Z. (220701236)

in partial fulfillment 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 **"FULL STACK PROJECT AUTOMATION BOT "** is the bonafide work of **"SAFEEKUR RAHMAN(220701236)"** who carried out the project work for the subject OAI1903 - Introduction to Robotic Process Automation under my supervision.

SIGNATURE

DR.N. DURAI MURUGAN,
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

"Full-Stack Project Setup Automation Bot" is an advanced Robotic Process Automation (RPA) tool engineered to simplify the initialization of full-stack projects in Visual Studio Code (VSCode) using UiPath. This intelligent bot leverages the Type Into, Use Application/Browser, Click Event, Hotkey Event, and Message Box functionalities to automate the setup process for projects involving React and Express. Users specify their desired project configurations, and the bot systematically executes the necessary commands and configurations within VSCode. By orchestrating a series of automated interactions, the bot ensures a swift and error-free setup, enhancing productivity and reducing manual effort in project initialization. Upon completion, the bot provides a confirmation message, indicating the successful establishment of the full-stack project environment. The **"Full-Stack Project Setup Automation Bot"** harnesses the power of RPA to streamline development workflows, ensuring consistency and efficiency in setting up React and Express projects.

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 **Mr. S. Meganathan, B.E, F.I.E.,** our Vice Chairman **Mr. Abhay Shankar Meganathan, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) Thangam Meganathan, 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 her guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guides, **Dr.N.Durai Murugan, M.E., ph.D.** Assistant Professor (SG), **Ms. Farjana, M.E.,** Assistant Professor (SG)**.** Rajalakshmi Engineering College 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.

Safeekur Rahman (220701236)

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**iii**
	LIST OF FIGURES	**vi**
	LIST OF ABBREVIATIONS	**vii**
1.	**INTRODUCTION**	**1**
	1.1 INTRODUCTION	1
	1.2 OBJECTIVE	3
	1.3 EXISTING SYSTEM	3
	1.4 PROPOSED SYSTEM	4
2.	**LITERATURE REVIEW**	**5**
3.	**SYSTEM DESIGN**	**9**
	3.1 SYSTEM FLOW DIAGRAM	9
	3.2 ARCHITECTURE DIAGRAM	10
	3.3 SEQUENCE DIAGRAM	11
4.	**PROJECT DESCRIPTION**	**12**
	4.1 MODULES	12
	4.1.1. INPUT HANDLING AND INITIALIZATION	12
	4.1.2. CONTENT ANALYSIS	12
	4.1.3. RESULT MANAGEMENT	13
	4.1.4. COMPLETION AND REPORTING	13
5.	**OUTPUT SCREENSHOTS**	**14**
6.	**CONCLUSION**	**18**
	APPENDIX	**19**
	REFERENCES	**25**

LIST OF FIGURES

Figure No.	Figure Name	Page No.
3.1	System Flow Diagram	9
3.2	Architecture Diagram	10
3.3	Sequence Diagram	11
5.1	Input Dialog	14
5.2	Excel Creation	14
5.3	AI Content Detection	15
5.4	Plagiarism Detection	16
5.5	Excel Report	17

LIST OF ABBREVIATIONS

ABBREVIATION	ACCRONYM
RPA	Robotic Process Automation
AI	Artificial Intelligence
API	Application Programming Interface
CV	Computer Vision
OCR	Optical Character Recognition

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

"Full-Stack Project Setup Automation Bot" is an innovative solution at the confluence of Robotic Process Automation (RPA) and software development efficiency. In the fast-paced realm of technology, the challenge of initializing full-stack projects swiftly and accurately is paramount. This cutting-edge bot, developed using UiPath, emerges as a comprehensive tool designed to automate the setup of full-stack projects involving React and Express within Visual Studio Code (VSCode).

For developers and teams navigating the complexities of project initialization, this bot offers a transformative approach. By automating commands and configurations, the bot not only accelerates the setup process but also enhances the consistency and accuracy of project environments. It addresses the evolving demands of development workflows, where the need for speed and precision in setting up development environments necessitates advanced, automated solutions.

UiPath's technology, known for automating repetitive digital tasks, combines AI computer vision, APIs, and access to pre-built automation components that seamlessly integrate into development processes. Historically, UiPath products have simplified tasks across various business software platforms, automating internal processes such as accounting and HR paperwork, and extending to customer management in the front office. UiPath's built-in OCR engines from Google, Microsoft, and ABBYY provide robust capabilities to read information from screens or scanned documents.

The primary product, UiPath Automation Platform, integrates a suite of low-code visual development tools called Studio for process creation, paired with Robots that execute these automated processes. The **"Full-Stack Project Setup Automation Bot"** leverages these powerful capabilities to streamline the establishment of React and Express projects, ensuring developers can focus more on coding and less on setup logistics.

1.2 OBJECTIVE

The primary objective of the **"Full-Stack Project Setup Automation Bot"** is to revolutionize the initial setup process for full-stack development projects. By leveraging Robotic Process Automation (RPA) through UiPath, the bot aims to automate the configuration of React and Express projects within Visual Studio Code (VSCode). This initiative seeks to provide developers and teams with an efficient and consistent solution to streamline project initialization.

1.3 EXISTING SYSTEM

In the current software development landscape, setting up full-stack projects is a manual and often repetitive process. Developers spend considerable time configuring project environments, installing dependencies, and setting up the necessary folders and files. This manual setup can lead to inconsistencies and errors, impacting productivity. The need for an automated and reliable solution to enhance the efficiency and accuracy of project setup is clear.

1.4 PROPOSED SYSTEM

The **"Full-Stack Project Setup Automation Bot"** is envisioned as a groundbreaking solution to address the current challenges in project initialization. Utilizing UiPath's robust

RPA capabilities, the bot automates the configuration of full-stack projects by executing a series of predefined tasks in VSCode. This includes setting up React and Express environments, installing necessary dependencies, and configuring project structures.

The proposed system aims to drastically reduce the manual effort required for project setup, providing developers with a tool that ensures swift, consistent, and accurate configurations. The bot's capabilities extend to automating key steps such as folder creation, dependency installation, and environment configuration. By offering a seamless setup process, the **"Full-Stack Project Setup Automation Bot"** empowers developers to focus more on coding and less on initial project logistics, thereby boosting productivity

CHAPTER 2 LITERATURE REVIEW

.1 Survey on Robotic Process Automation (RPA) in Software Development:

Robotic Process Automation (RPA) is increasingly recognized as a valuable asset in software development, automating repetitive tasks to boost efficiency and reduce manual effort. For example, RPA has been successfully implemented in automating build processes, code reviews, and deployment pipelines, significantly easing the workload of developers. However, challenges persist, particularly in automating complex decision-making and tasks that require human creativity. The literature review of research papers related to RPA in Software Development is listed below:

[1] A study discusses how the integration of RPA in software development can lead to substantial time savings and reduced error rates. The research highlights the automation of repetitive coding tasks and its impact on improving developer productivity.

[2] Another research paper from the International Journal of Computer Applications (IJCA) explores the application of RPA in continuous integration and continuous deployment (CI/CD) pipelines. The paper concludes that RPA can streamline these processes, resulting in faster and more reliable software releases.

2.2 Survey on Full-Stack Development Automation:

Automating full-stack development tasks is an area of growing interest, particularly in the context of enhancing productivity and maintaining consistency. Existing tools and frameworks offer varying degrees of automation, from scaffolding project structures to

managing dependencies. The literature review of research papers related to Full-Stack Development Automation is listed below:

[3] Research investigates the use of automation tools for setting up full-stack development environments. The study highlights the benefits of automating project initialization, including reduced setup time and improved consistency across development teams.

[4] A paper from the Journal of Software Engineering examines the effectiveness of automation frameworks in managing full-stack development workflows. The research concludes that automation can significantly reduce the manual effort involved in environment setup and configuration.

2.3 Survey on Efficiency in Software Project Initialization:

Efficiency in software project initialization is critical to kick-starting development activities smoothly. Automation plays a key role in achieving this efficiency by minimizing the time and effort required for initial setup. The literature review of research papers related to efficiency in software project initialization is listed below:

[5] An article discusses various automation techniques for initializing software projects, focusing on the integration of RPA tools to streamline initial setup processes. The study emphasizes the time-saving benefits and reduction in human errors.

[6] A survey by the Institute of Electrical and Electronics Engineers (IEEE) provides insights into different approaches to automate project setup, including the use of scripting and RPA. The paper underscores the potential of these methods in enhancing setup efficiency and consistency.

2.4 Summary of the Intersection of RPA, Full-Stack Automation, and Project Initialization Efficiency:

The "Full-Stack Project Setup Automation Bot" integrates these areas, leveraging RPA to automate the initialization of full-stack projects involving React and Express. The project incorporates advanced automation techniques to execute necessary commands and configurations within VSCode. Additionally, it addresses the challenges of maintaining consistency and reducing setup time, positioning the bot as a pivotal tool in enhancing development workflows.

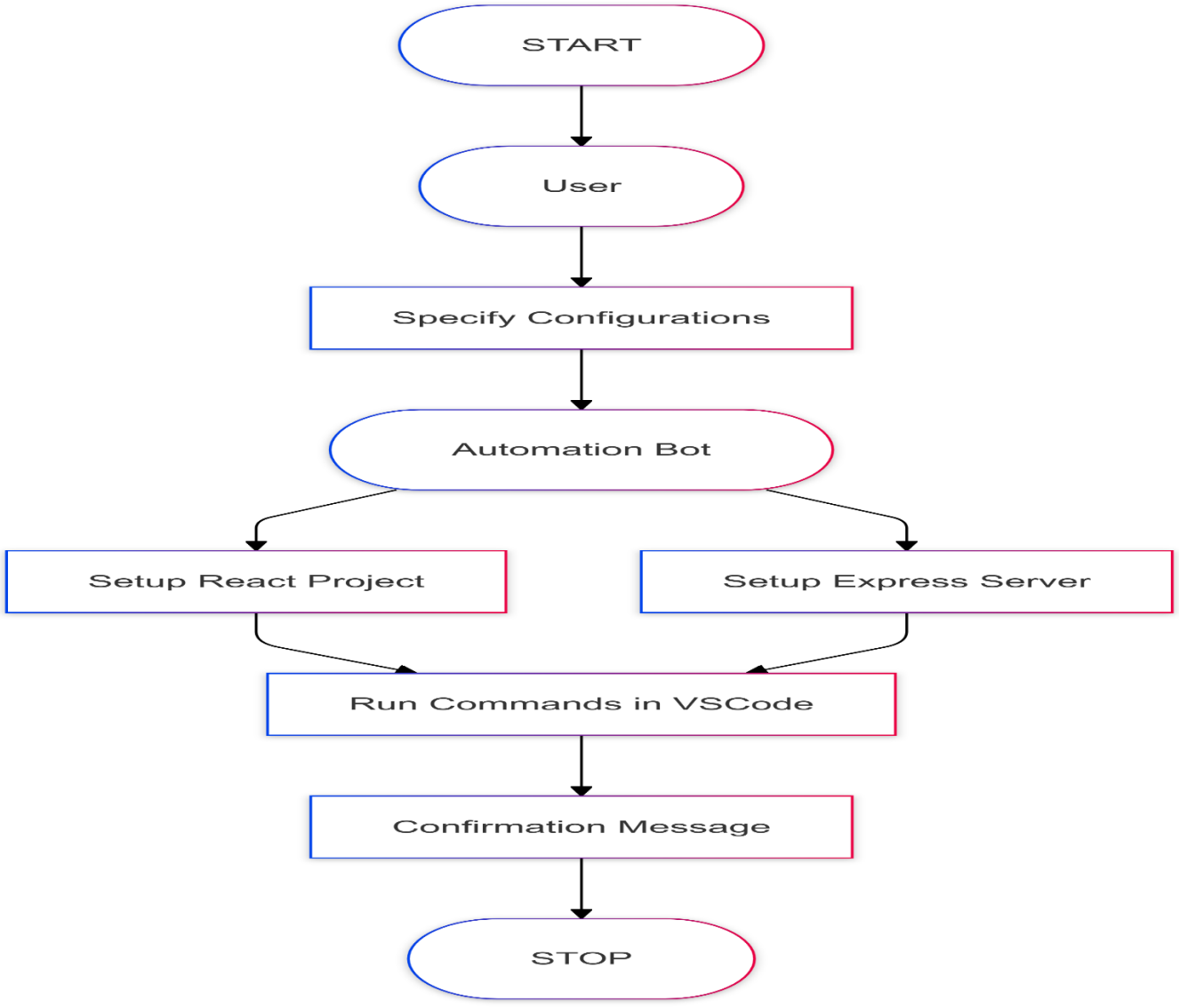
The project’s innovative approach aligns with current trends in software development, presenting a timely solution to the complexities of project setup. The integration of RPA with full-stack development automation showcases the bot's significance in contributing to more efficient and reliable software development processes.

CHAPTER 3

SYSTEM DESIGN

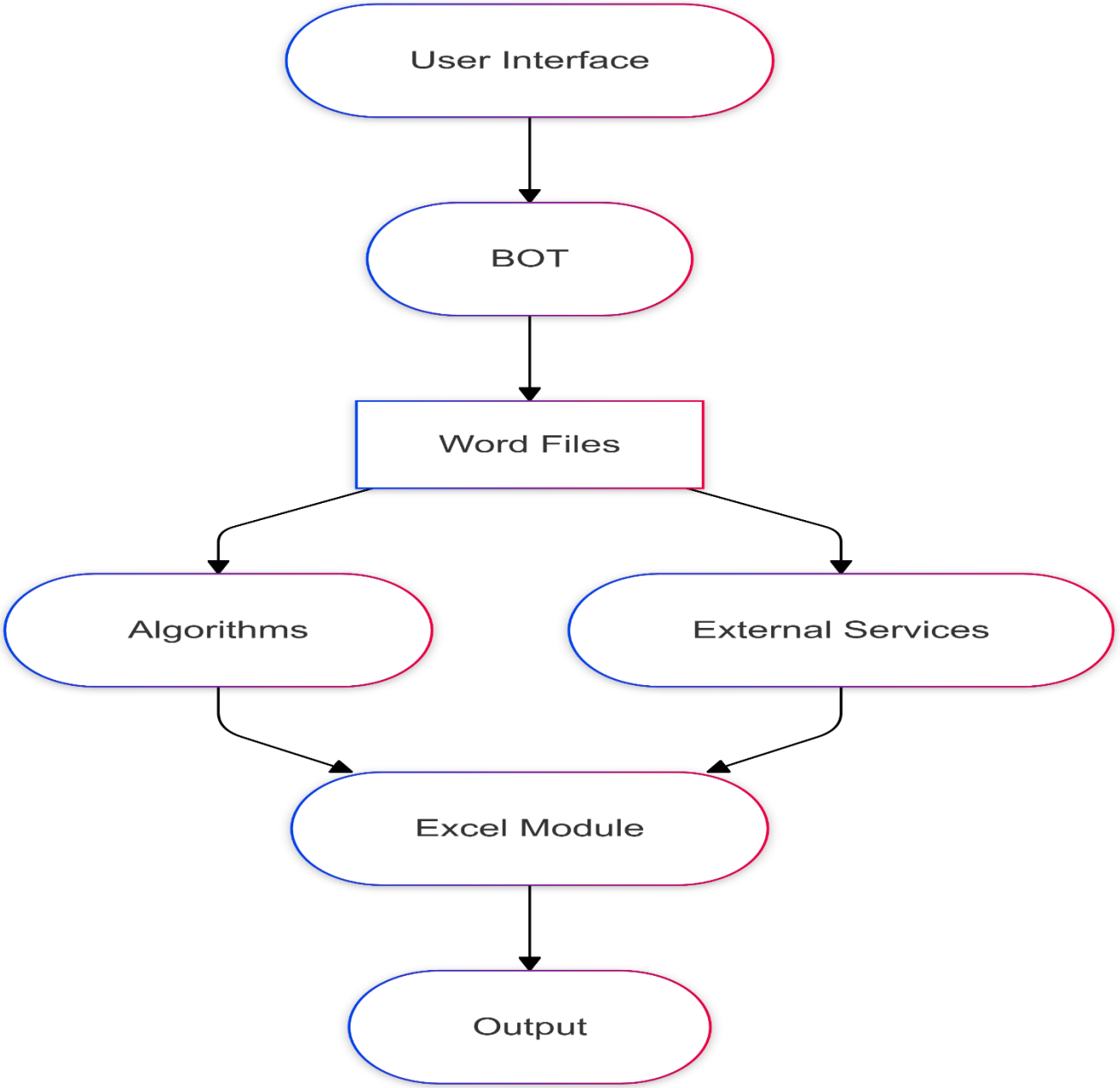
3.1 SYSTEM FLOW DIAGRAM

A flowchart is a type of diagram that represents an algorithm, workflow or process. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem.



3.2 ARCHITECTURE DIAGRAM

An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components. The architecture diagram for this project is in Fig. 3.2.



3.3 SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram because it describe and s how in what order a group of objects works together. The sequence diagram for this project is in Fig. 3.3.

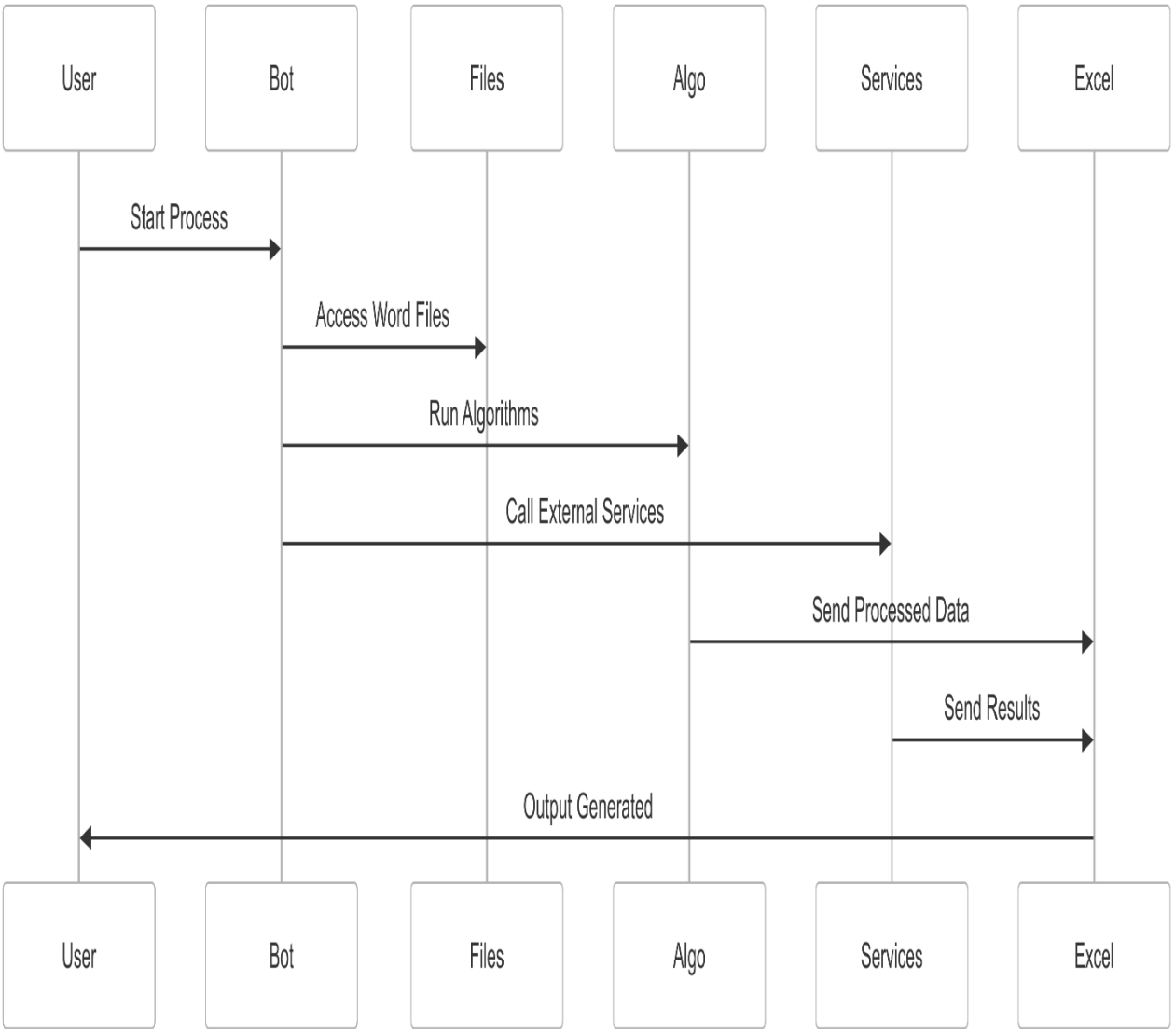


Fig 3.3 Sequence Diagram

CHATPTER 4 PROJECT DESCRIPTION

Project Description: Word File Data Processing Automation Bot
The project focuses on developing an **automation bot** designed to streamline the processing of Word files and generate meaningful outputs in Excel format. This system integrates multiple components to efficiently handle input files, apply algorithms for data processing, and incorporate external services when required, ensuring seamless and accurate result generation.

Key Features:

1. **User Interface**:
   - Provides an interface for users to initiate the process and manage inputs.
   - Simple and intuitive interaction with the bot.

2. **Automation Bot**:
   - Automates the workflow, eliminating the need for manual intervention.
   - Accesses Word files and orchestrates the subsequent data processing steps.

3. **Word Files Access**:
   - Reads and extracts data from Word files in the selected directory.
   - Ensures compatibility with common Word document formats.

4. **Algorithms**:
   - Processes the extracted data using custom algorithms to evaluate or transform the content.
   - Optimized for accurate and efficient data processing.

5. **External Services**:
   - Utilizes third-party APIs or tools to enhance data processing, such

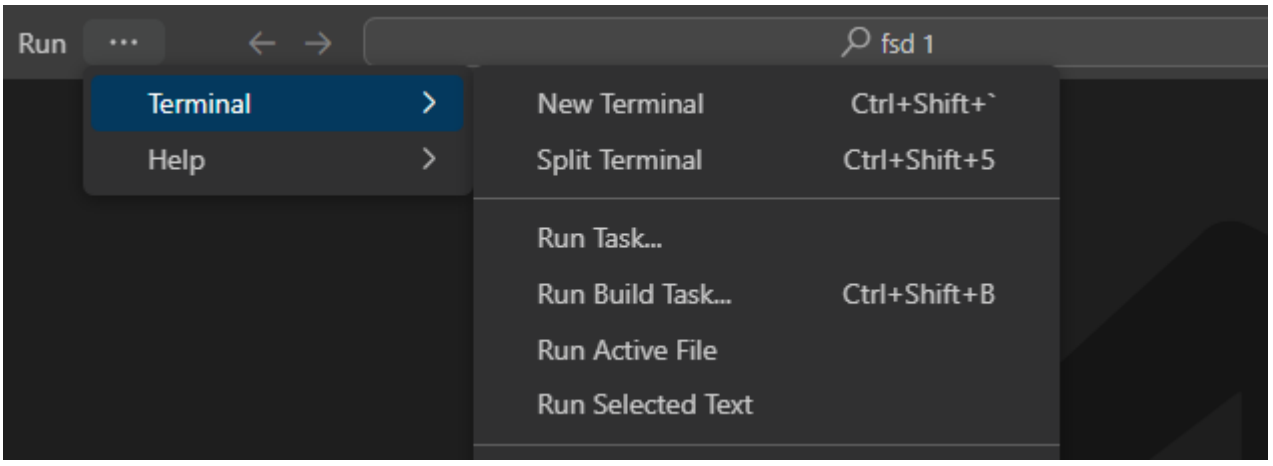
as validation, enrichment, or analysis.
    - Ensures reliable integration with external platforms.

6. **Excel Module**:
    - Combines processed results into a structured Excel sheet.
    - Prepares the final output in a format suitable for analysis or reporting.

7. **Output Generation**:
    - Produces the final output file for the user.
    - Ensures accurate, readable, and actionable results.

CHAPTER 5

OUTPUT SCREENSHOTS



Creating client project

```
found 0 vulnerabilities

C:\Users\saf_rah_a.z\OneDrive\Desktop\fsd 1>mkdir client
```

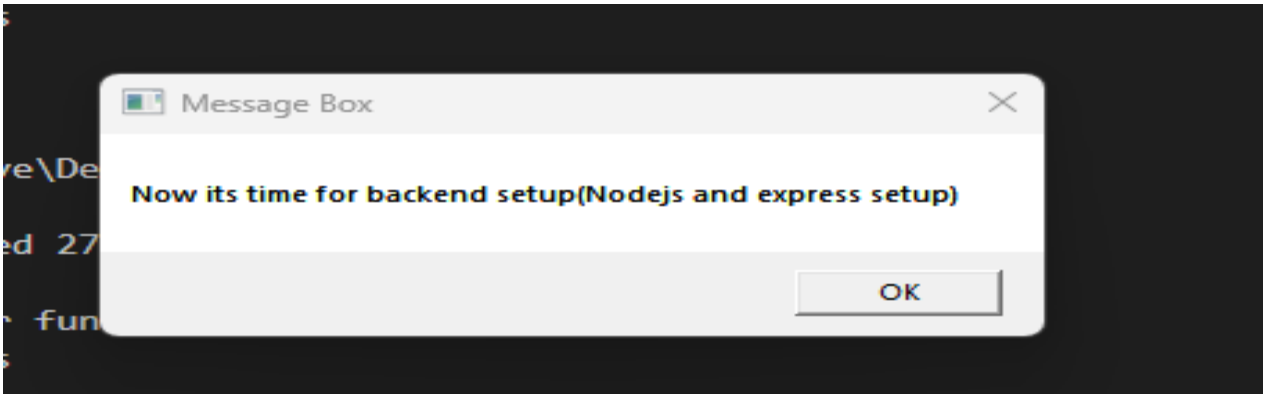
Set upping react

```
C:\Users\saf_rah_a.z\OneDrive\Desktop\fsd 1>npm create vite
√ Project name: ... client
? Select a framework: » - Use arrow-keys. Return to submit.
>   Vanilla
    Vue
    React
    Preact
    Lit
    Svelte
    Solid
```

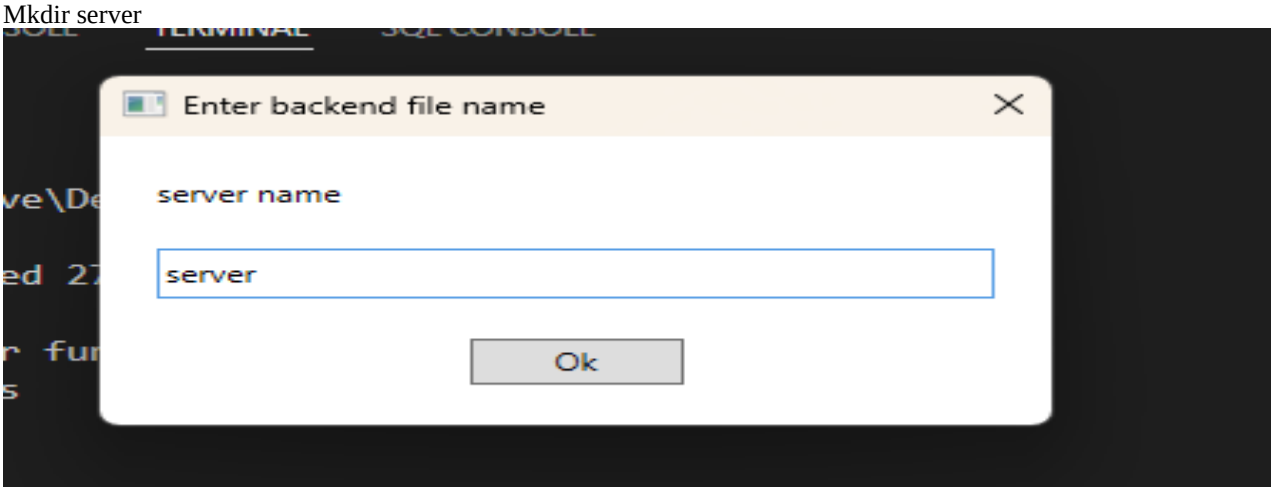
```
C:\Users\saf_rah_a.z\OneDrive\Desktop\fsd 1\client>npm instal
Terminate batch job (Y/N)? y

C:\Users\saf_rah_a.z\OneDrive\Desktop\fsd 1\client>npm install
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache
if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and power
ful.
npm WARN deprecated @humanwhocodes/config-array@0.13.0: Use @eslint/config-array instead
npm WARN deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated @humanwhocodes/object-schema@2.0.3: Use @eslint/object-schema instead
[##################] \ reify:@babel/helpers: timing reifyNode:node_modules/@babel/types Completed in 2816ms
```

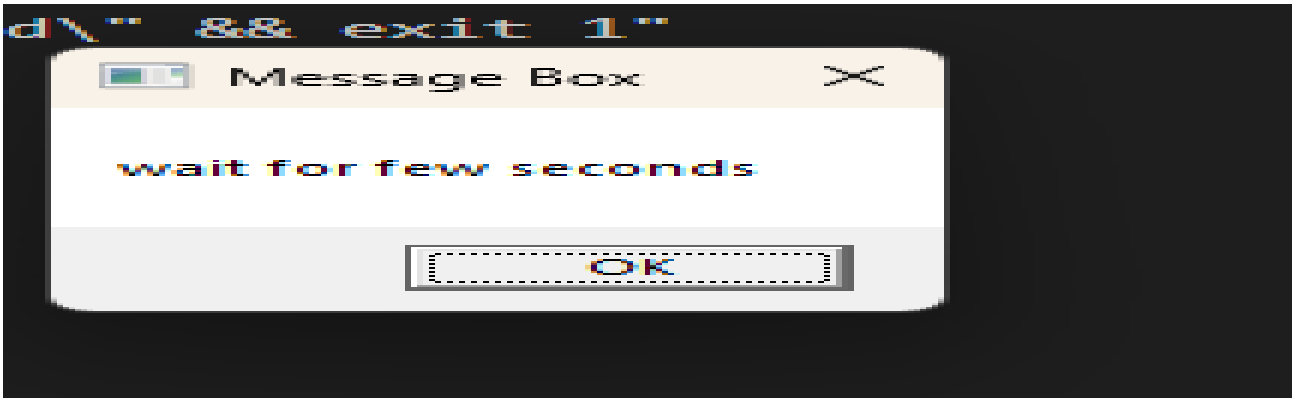
Set upping backend server

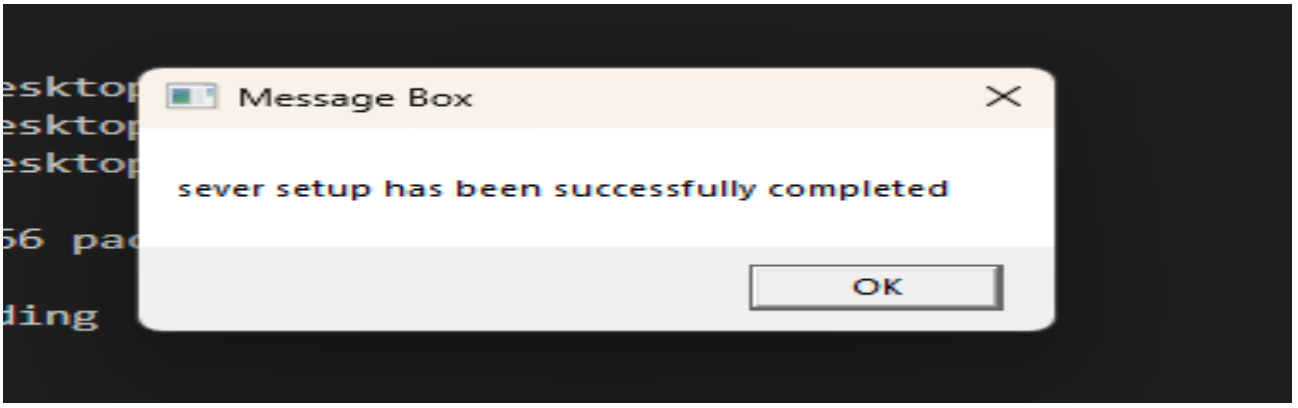
```
C:\Users\saf_rah_a.z\OneDrive\Desktop\fsd 1>mkdir server

C:\Users\saf_rah_a.z\OneDrive\Desktop\fsd 1>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (fsd-1)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
```



Mkdir server







CHAPTER 6

CONCLUSION

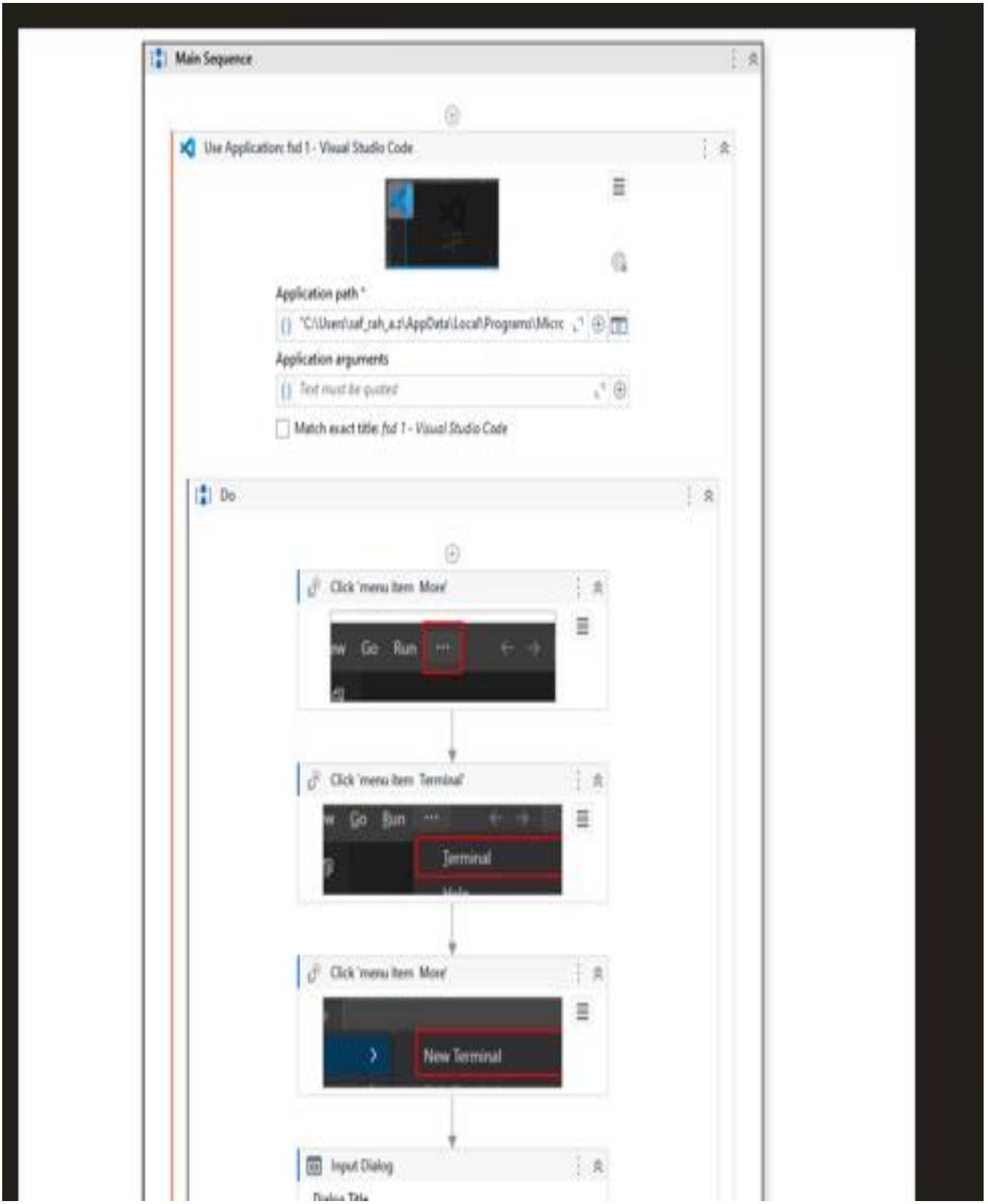
The **"Full-Stack Project Setup Automation Bot"** significantly enhances the efficiency and accuracy of initializing full-stack development projects. By leveraging UiPath's Robotic Process Automation (RPA), the bot automates the setup of React and Express environments within Visual Studio Code (VSCode). This innovative solution reduces the manual effort involved in project configuration, ensuring consistency and reliability across development teams.

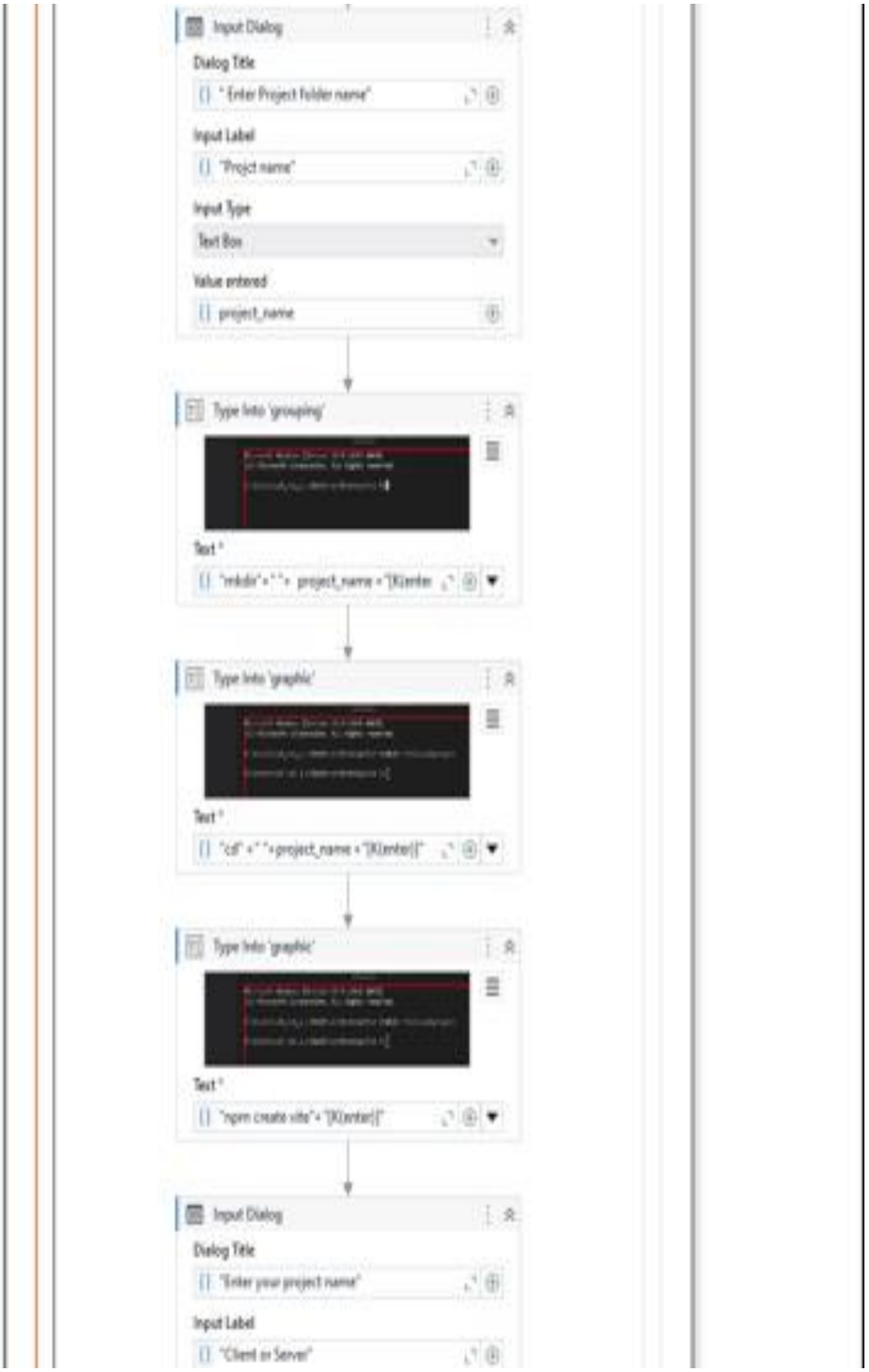
The bot's ability to provide real-time progress updates enhances transparency throughout the setup process, offering developers a seamless and user-friendly experience. By automating repetitive and error-prone tasks, the bot allows developers to focus on more complex and creative aspects of their work, ultimately boosting productivity.

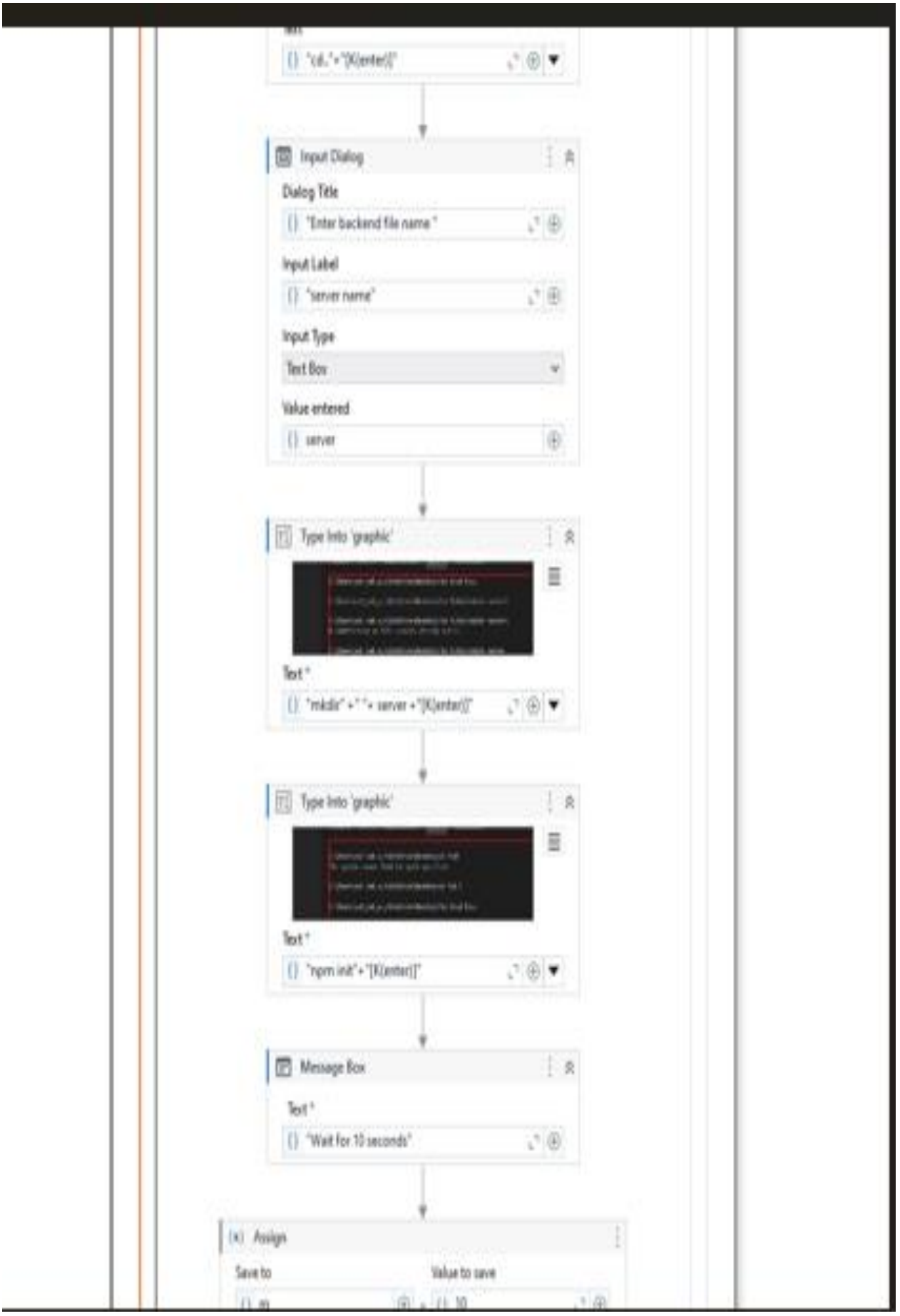
While the bot excels in automating setup processes, ongoing updates and improvements are essential to adapt to evolving development practices and tools. The successful implementation of the **"Full-Stack Project Setup Automation Bot"** represents a significant step forward in optimizing development workflows, contributing to more efficient and streamlined project initiation in the software industry.

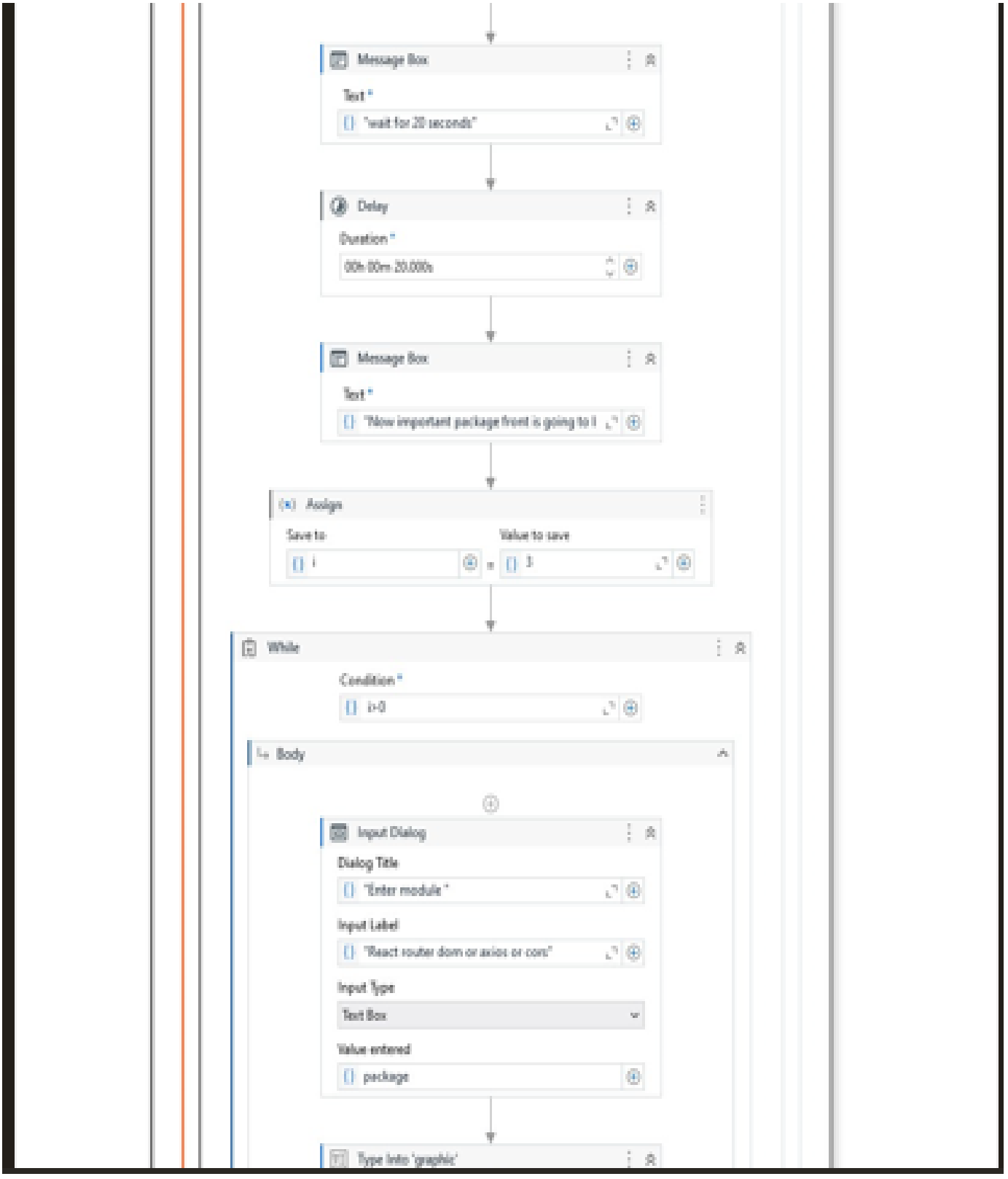
APPENDIX

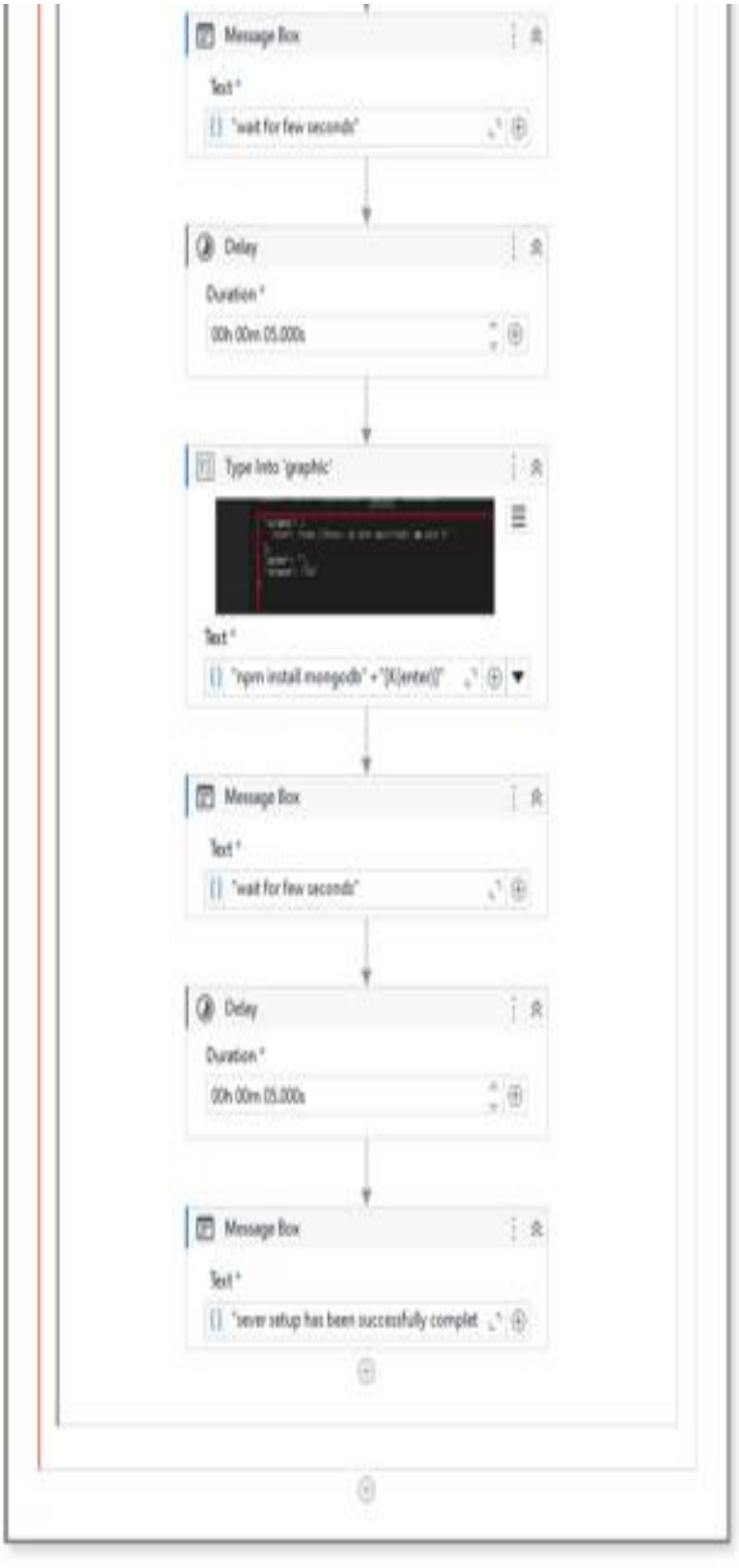
PROCESS WORK FLOW




Input Dialog
Dialog Title
" Enter Project Folder name"
Input Label
"Project name"
Input Type
Text Box
Value entered
project_name
Type Into 'grouping'
Text "
"mkdir"+" "+ project_name + "[Enter
Type Into 'graphic'
Text "
"cd" +" "+project_name + "[Enter]["
Type Into 'graphic'
Text "
"npm create vite"+ "[Enter]["
Input Dialog
Dialog Title
"Enter your project name"
Input Label
"Client or Server"


"cd.." + "{K}enter}{}"
Input Dialog
Dialog Title
"Enter backend file name "
Input Label
"server name"
Input Type
Text Box
Value entered
server
Type Into 'graphic'
Text *
"mkdir" + " " + server + "{K}enter}{}"
Type Into 'graphic'
Text *
"npm init" + "{K}enter}{}"
Message Box
Text *
"Wait for 10 seconds"
Assign
Save to
Value to save
10


Message Box
Text *
"wait for 20 seconds"
Delay
Duration *
00h 00m 20.000s
Message Box
Text *
"Now important package front is going to l
Assign
Save to
i
Value to save
3
While
Condition *
i>0
Body
Input Dialog
Dialog Title
"Enter module"
Input Label
"React router dom or axios or cors"
Input Type
Text Box
Value entered
package
Type into 'graphic'


Message Box
Text *
"wait for few seconds"
Delay
Duration *
00h 00m 05.000s
Type Into 'graphic'
Text *
"npm install mongodb" + "{Enter}{"
Message Box
Text *
"wait for few seconds"
Delay
Duration *
00h 00m 05.000s
Message Box
Text *
"sever setup has been successfully complet

REFERENCES

[1] Kuppusamy, Palanivel & Joseph K, Suresh. (2020). Robotic Process Automation to Smart Education. 3775.

[2] Patil, Dr & Mane, Vinod & Patil, Dr. (2019). Social Innovation in Education System by using Robotic Process Automation (Rpa). International Journal of Innovative Technology and Exploring Engineering. 8. 3757-3760. 10.35940/ijitee.K2148.0981119.

[3] Elkhatat, A.M., Elsaid, K. & Almeer, S. Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text. *Int J Educ Integr* **19**, 17 (2023). https://doi.org/10.1007/s40979-023-00140-5

[4] H. Alamleh, A. A. S. AlQahtani and A. ElSaid, "Distinguishing Human-Written and ChatGPT-Generated Text Using Machine Learning," 2023 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2023, pp. 154-158, doi: 10.1109/SIEDS58326.2023.10137767.

[5] Tomáš Foltýnek, Norman Meuschke, and Bela Gipp. 2019. Academic Plagiarism Detection: A Systematic Literature Review. ACM Comput. Surv. 52, 6, Article 112 (November 2020), 42 pages. https://doi.org/10.1145/3345317

[6] H. A. Chowdhury, D. K. Bhattacharyya, "Plagiarism: Taxonomy, Tools and Detection Techniques", 19th National Convention on Knowledge, Library and Information Networking, 2018.