

# **DIGITAL JOURNAL AND TASKS AUTOMATION**

**A PROJECT REPORT**

*Submitted by*

**JAYAJOTHI KUMAR (220701100)**

*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 “**DIGITAL JOURNAL AND TASKS AUTOMATION**” is the bonafide work of “**JAYAJOTHI KUMAR (220701100)**” who carried out the project work for the subject OAI1903-Introduction to Robotic Process Automation under my supervision.

Mrs. J. Jinu Sophia (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 \_\_\_\_\_.

# TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1.	ABSTRACT	5
2.	INTRODUCTION	6
3.	OBJECTIVE	7
4.	EXISTING SYSTEM	8
5.	PROPOSED SYSTEM	9
6.	SYSTEM REQUIERMENTS	11
	6.1 SOFTWARE REQUIREMENTS	11
	6.2 HARDWARE REQUIREMENTS	12
7.	PROJECT DESCRIPTION	13
8.	SYSTEM DESIGN	15
	8.1 ARCHITECTURE	15
	8.2 SEQUENCE DIAGRAM	16
9.	PROCESS DESIGN	17
	9.1 MAIN PROCESS	17
	9.2 SUB-PROCESS	18
10.	IMPLEMENTATION	19
11.	TESTING	28
12.	CONCLUSION	30
13.	FUTURE ENHANCEMENT	31
14.	REFERENCES	32

## 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.

**Jayajothi Kumar (220701100)**

## **ABSTRACT**

This project automates the management of a React-based digital journal using UiPath, streamlining repetitive tasks like server setup, data extraction, and task management. The system automates the initialization of the XAMPP server, including starting Apache and MySQL, and launches the React application. Journal entries are extracted from PDFs, processed, and added to the journal, while task details from Excel files are integrated seamlessly through web automation.

Key features include user prompts for controlling application shutdown, allowing the system to stop XAMPP and close related processes upon completion. This approach reduces manual effort, minimizes errors, and improves productivity. With a modular design, the system is scalable for future enhancements, such as cloud integration or advanced analytics. This project provides an efficient, user-friendly solution for managing digital journaling tasks through automation.

# INTRODUCTION

In today's fast-paced digital world, the need for efficient data management and automation is more critical than ever. Repetitive tasks such as data entry, system setup, and task management can consume significant time and lead to errors, reducing overall productivity. This project aims to address these challenges by automating the processes involved in managing a React-based digital journal application using UiPath, a powerful Robotic Process Automation (RPA) tool.

The digital journal system serves as an online platform where users can manage personal journal entries and tasks. However, the process of updating entries, extracting data from external sources like PDFs and Excel files, and managing server configurations can be tedious and error-prone. By implementing UiPath automation, these tasks are streamlined, significantly reducing manual effort and minimizing the chances of mistakes.

The automation begins with starting the XAMPP server (Apache and MySQL), which is required to run the React application. Once the server is running, the system automatically extracts data from PDFs and Excel files, processes it, and inputs the relevant details into the journal through web automation. Additionally, users can manage tasks, including adding and prioritizing them, directly from Excel. At the end of the session, the system prompts users to confirm whether they wish to shut down all applications, providing a clean and efficient shutdown process.

Through this approach, users can focus more on creative and productive tasks, while the system handles routine operations.

## **OBJECTIVE**

The primary objective of this project is to automate the management of a React-based digital journal application using UiPath, improving productivity and minimizing human error by automating repetitive tasks. This allows users to focus on more important tasks while ensuring efficient and accurate data handling.

The first objective is to automate the setup of the XAMPP server, including starting Apache and MySQL services, and launching the React journal application. This reduces the need for manual configuration and ensures the system is ready for use quickly.

Next, the system automates the extraction of journal entry details from PDFs and inputs them into the digital journal through web automation, improving the speed and accuracy of data entry.

Another goal is to automate task management by extracting task data from Excel files and adding it to the journal application, simplifying task tracking and updates.

Finally, the system includes a user-controlled shutdown feature, prompting the user to confirm if they want to close all running applications, including the XAMPP server and command prompt, for a clean shutdown.

Overall, the project aims to create a seamless automation solution for managing journal entries and tasks, showcasing the efficiency and potential of Robotic Process Automation (RPA) in everyday applications.

## **EXISTING SYSTEM**

In the current scenario, managing journal entries and tasks is often a manual and time-consuming process. Users must rely on standalone tools or manual data entry methods, which can lead to inefficiencies, errors, and lack of integration between different components of the workflow. Below are the key limitations of the existing system:

### **1. Manual Server and Application Setup**

- Users need to manually start the XAMPP server, open command prompts, and launch the React-based journal application each time they wish to use the system. This process is repetitive and prone to errors if not done correctly.

### **2. Journal Entry Management**

- Journal entries from PDFs or other documents must be manually extracted and entered into the system. This requires significant time and effort, especially when dealing with large volumes of data.

### **3. Task Management**

- Tasks are typically maintained in Excel sheets or similar tools, requiring users to switch between applications to manually input tasks into the journal system. This process increases the risk of data duplication or loss.



## **PROPOSED SYSTEM**

The proposed system automates the management of a React-based digital journal application using UiPath, aiming to simplify repetitive tasks, improve accuracy, and enhance overall productivity. This system will address the common challenges of manually managing journal entries and tasks, particularly in scenarios where large amounts of data need to be processed regularly.

At the core of the proposed system is the automation of server initialization and React application startup. The system uses UiPath to automatically launch the XAMPP server (starting Apache and MySQL) and the React application via the command prompt, reducing the manual effort required for setup.

Another key feature is the automation of journal entry creation. The system extracts text from PDF files containing journal entry details using UiPath's PDF activities, processes this data, and inputs it into the journal website using web automation. This ensures that data is entered accurately and efficiently without requiring manual typing.

For task management, the system extracts task details from Excel files, including task names, priority levels, and due dates, and automatically updates the journal application with this information. This reduces the need for users to manually enter task details, improving task tracking and management.

The system also includes a user prompt to confirm whether they want to close all running applications at the end of the process. If the user confirms, the system will stop the XAMPP server, close command prompt windows, and cleanly exit all applications.

Overall, the proposed system aims to enhance the workflow by automating key operations, improving efficiency, reducing human error, and providing users with more time to focus on creative and productive tasks. This system demonstrates the value of Robotic Process Automation (RPA) in automating data-driven tasks and streamlining digital journal management.

# SYSTEM REQUIREMENTS

## Software Requirements

### 1. UiPath Studio:

- The primary tool used for automation. It is used to design, develop, and deploy RPA workflows to automate the tasks such as server startup, data extraction, and web automation.

### 2. XAMPP:

- A software package containing Apache, MySQL, and PHP, used to set up and manage the local server environment for running the React-based journal application.

### 3. React:

- A JavaScript library for building user interfaces, used for creating the digital journal application. The journal will be hosted and run through the local server created by XAMPP.

### 4. Node.js:

- JavaScript runtime used to run the React application on the local server, ensuring the journal's front-end application works as intended.

### 5. Microsoft Excel:

- Used for task management, where task details are stored in an Excel file. UiPath uses Excel automation to read and update the tasks in the journal system.

### 6. Web Browser (Google Chrome):

- A web browser is necessary to interact with the React-based digital journal, as UiPath will use browser automation to input data into the journal application.

## Hardware Requirements

### 1. Processor:

- A minimum of **Intel Core i3** or equivalent processor to handle the automation tasks and ensure smooth performance while running the system.

### 2. RAM:

- **4 GB or more** of RAM to support UiPath Studio, XAMPP, Node.js, and other necessary applications running simultaneously without lag.

### 3. Storage:

- At least **10 GB of free disk space** for installing the required software (UiPath Studio, XAMPP, React, and Node.js) and storing project files, logs, and data.

### 4. Operating System:

- **Windows 10 or higher** is recommended for compatibility with UiPath Studio, XAMPP, and other necessary applications.

### 5. Internet Connection:

- A stable internet connection for downloading required software packages, libraries, and for accessing cloud services (if needed for future enhancements).

## PROJECT DESCRIPTION

This project focuses on automating the management of a React-based digital journal application using UiPath, an advanced Robotic Process Automation (RPA) tool. The system is designed to automate several tasks within the journal application, including server setup, data entry, and task management, ultimately improving productivity, reducing manual effort, and ensuring accuracy in managing journal entries and tasks.

### Core Features and Workflow

The project aims to simplify the process of managing a digital journal by automating several repetitive and error-prone tasks. The workflow begins by automating the initialization of the required server environment using XAMPP. This includes starting Apache and MySQL services and launching the React journal application. Once the server is up and running, the system proceeds to automate the process of adding journal entries and tasks.

1. **System Setup Automation:** The first task in the process involves the automation of setting up the XAMPP server. Using UiPath, the system will open the XAMPP control panel, start the Apache and MySQL services, and launch the React application via the command prompt. This eliminates the need for manual configuration and reduces the risk of errors in server setup.
2. **Journal Entry Automation:** Journal entries are extracted from PDF files using UiPath's PDF activities. The extracted data, including the title, content, and date, is processed using string manipulation techniques to ensure that the data is in the correct format before being entered into the journal application. Using UiPath's web automation features (such as **Type Into**, **Click**, and **Select Item**), the journal entry details are then added to the website.

3. **Task Management Automation:** Task details are stored in an Excel file and include information like task name, priority, and due date. The system extracts this information using UiPath's Excel automation tools, iterating through each task and adding it to the journal application. Tasks are updated through web automation, ensuring that the journal's task list is always up-to-date.
4. **User-Controlled Shutdown:** Once all tasks are completed, the system prompts the user to decide whether they want to shut down the applications. If confirmed, the system will stop the XAMPP server, close the command prompt, and terminate all associated processes, ensuring a clean and efficient shutdown of the system.

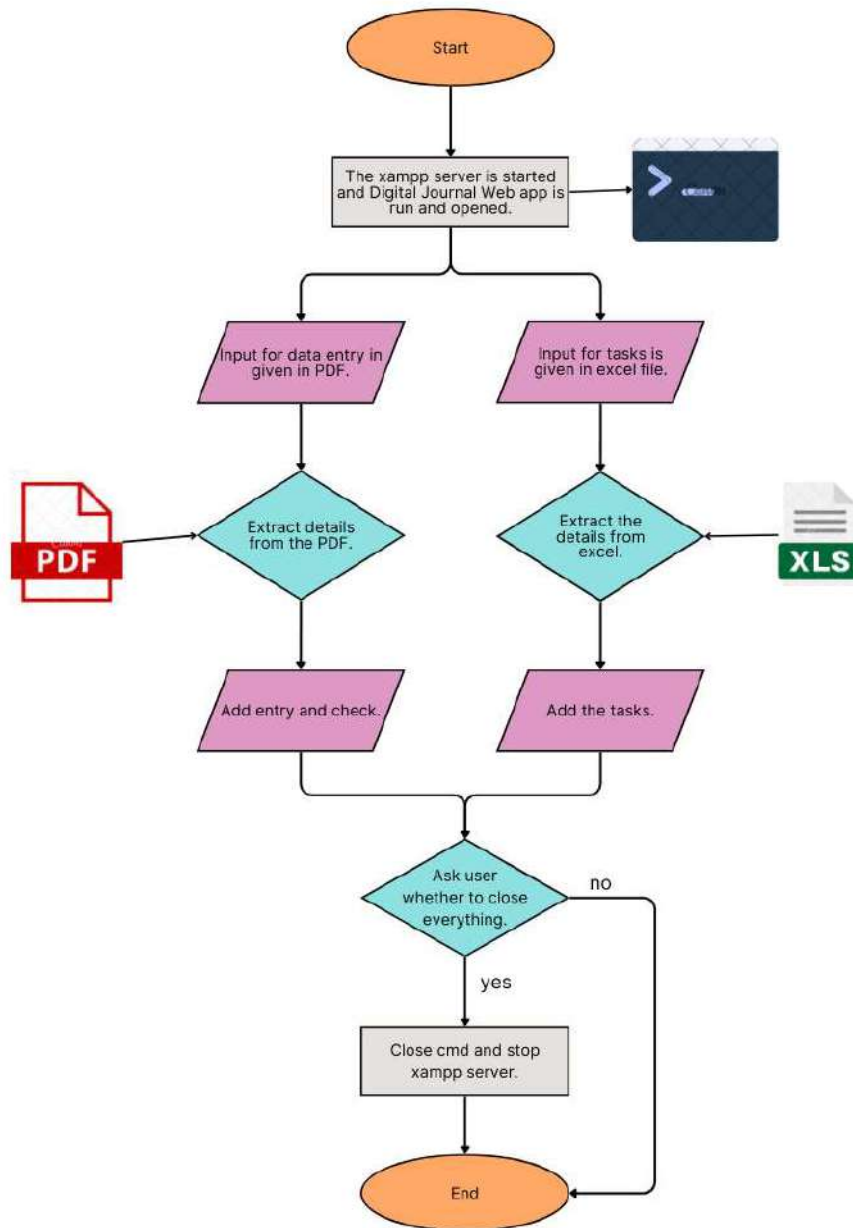
### **Benefits and Impact**

This automated system significantly reduces the manual effort involved in managing journal entries and tasks. It eliminates the need for users to manually start the server, enter journal details, or update tasks, thus improving efficiency and reducing human error. The automation also provides a clean and controlled shutdown, ensuring that all processes are closed properly.

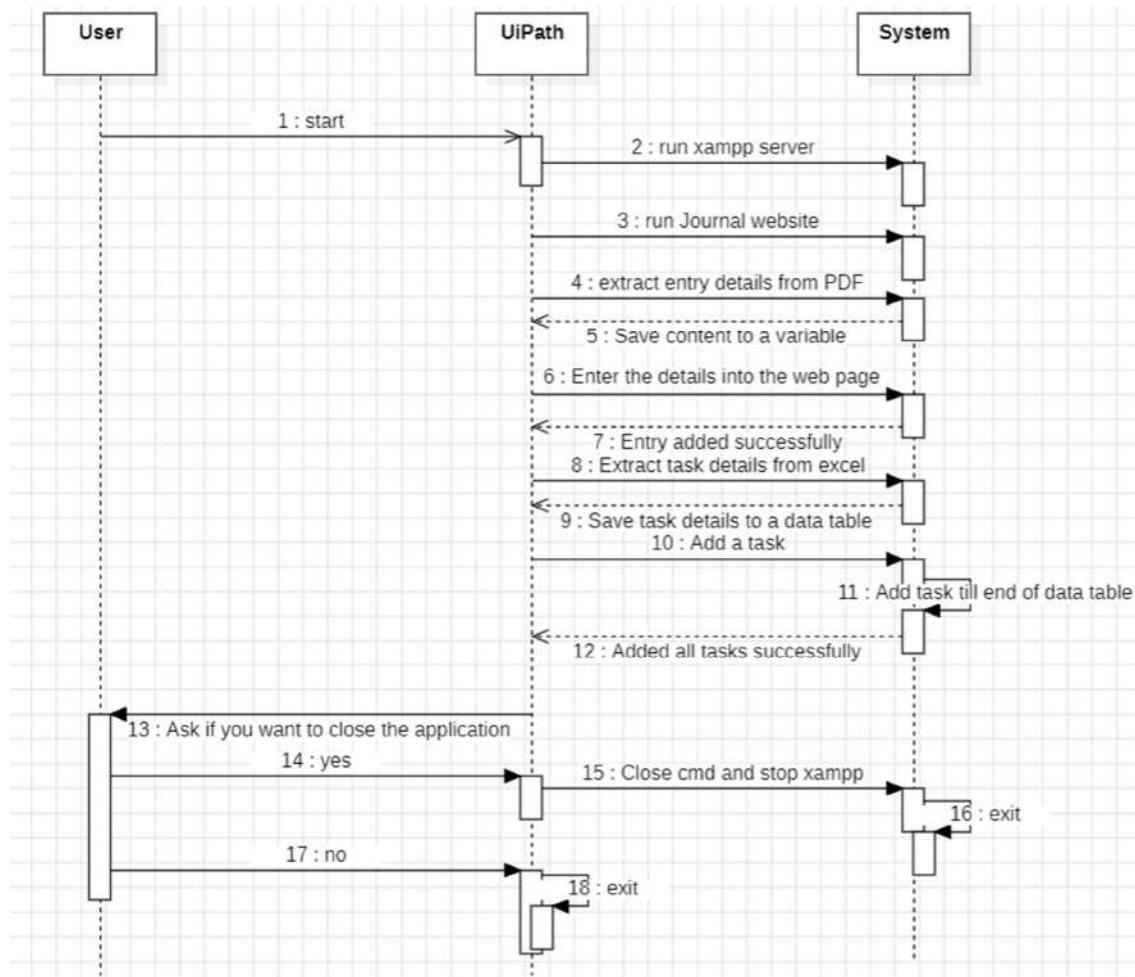
By implementing this system, users will benefit from a streamlined workflow that allows them to focus on more productive and creative tasks while the automation handles repetitive and time-consuming activities. This project demonstrates the power of Robotic Process Automation (RPA) to enhance the functionality of a digital journal application and can serve as a foundation for further enhancements, such as cloud integration or more advanced data processing features.

# SYSTEM DESIGN

## ARCHITECTURE



## SEQUENCE DIAGRAM





## PROCESS DESIGN

The process design for the proposed system focuses on creating an efficient and automated workflow for managing a React-based digital journal. The overall objective is to streamline the tasks involved in server setup, data entry, and task management by utilizing UiPath's RPA capabilities. Below, we outline the main processes and sub-processes involved in the automation system.

### MAIN PROCESS

The main process involves the initialization of the system, followed by the automation of journal entry creation and task management. The workflow is broken down into clear steps to ensure smooth execution.

#### 1. Server Setup and React Application Launch

- **Sub-process:** Using UiPath, the automation opens the XAMPP control panel and starts the Apache and MySQL services to run the local server.
- The React journal application is launched using the command prompt.

#### 2. Journal Entry Automation

- **Sub-process:** The system extracts data from PDFs (which contain the journal entry details) using UiPath's PDF activities. The extracted text is processed and formatted using string manipulation activities.
- The formatted journal entry is then entered into the React-based journal via web automation (using browser activities such as **Type Into** and **Click**).

### 3. Task Management Automation

- **Sub-process:** Task details are extracted from an Excel file using UiPath's Excel automation capabilities.
- The extracted tasks (task name, priority, due date) are added to the journal system using web automation, updating the task list in real time.

### 4. User-Controlled Shutdown

- **Sub-process:** After completing the journal entries and task updates, the system prompts the user to confirm if they would like to close all applications.
- If the user confirms, the system stops the XAMPP server, closes the command prompt, and terminates any other processes running on the system.

## SUB-PROCESSES

- **Server Setup (Sub-process):** The XAMPP control panel is launched via UiPath, and services such as Apache and MySQL are started. This process ensures that the React application can function correctly on the local server. If any service fails to start, the automation triggers an alert or error message.
- **Extracting and Formatting Journal Entries (Sub-process):** UiPath reads the journal entry PDF using its PDF activities and extracts the text. The extracted content undergoes string manipulation to structure the data, making it ready to input into the React application's entry fields. A series of validation checks are performed to ensure the extracted data is correct.
- **Updating Tasks from Excel (Sub-process):** The Excel automation sub-process retrieves task details, looping through each row to extract task data. String manipulation is applied to ensure correct formatting before adding it to the journal application through the browser automation activities.

- **Shutdown Process (Sub-process):** After completing all automation tasks, the user is prompted via a message box to decide whether to shut down the system. If the user confirms, the system will automatically close all processes, stop the XAMPP server, and cleanly shut down the system.

### **Flow Summary**

The process begins by automating the server startup and launching the React application. It then moves on to the extraction of journal entries from PDFs, followed by updating tasks from Excel. The system concludes by giving the user the option to shut down all applications. The entire process is automated and runs seamlessly, ensuring minimal user intervention.

This process design optimizes workflow efficiency and ensures that each task is automated to perfection, resulting in improved productivity and reduced manual effort.

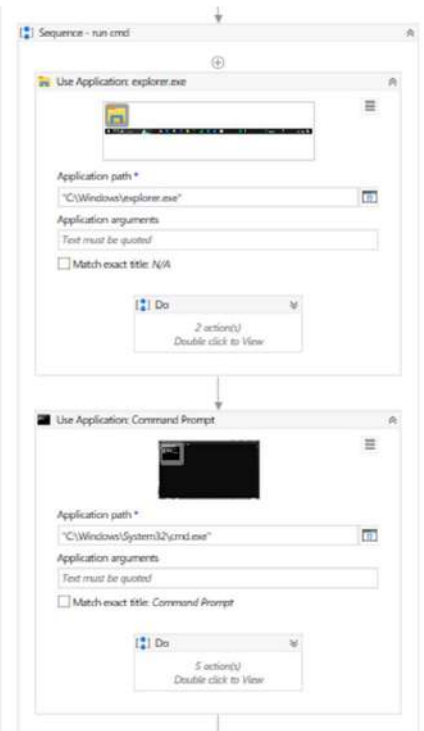
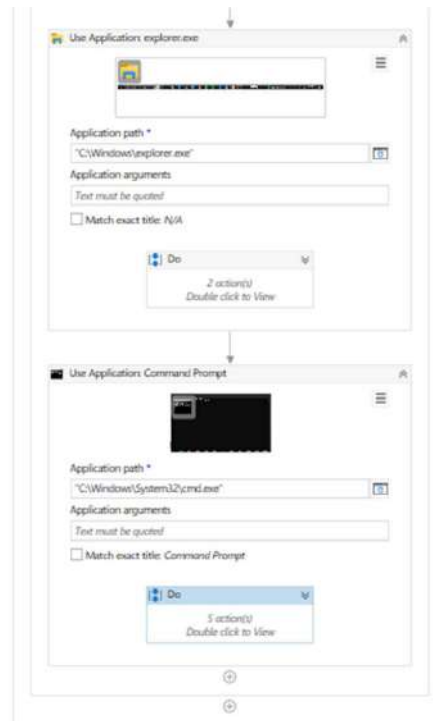
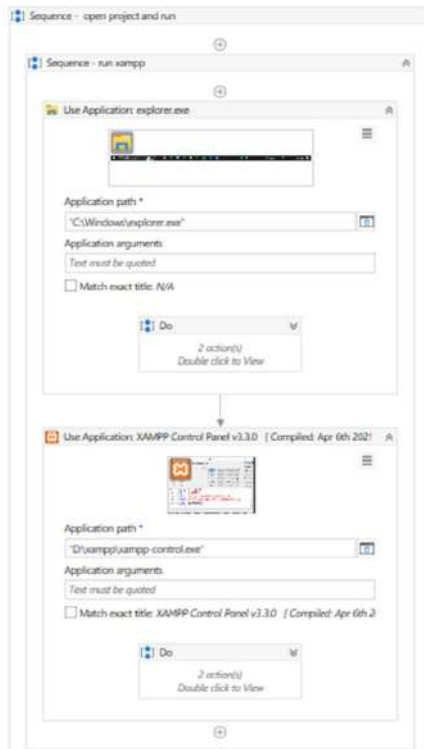
# IMPLEMENTATION

The implementation phase involves translating the designed processes into executable automation workflows using UiPath. The objective is to develop an efficient, error-free system that performs all the required tasks—server setup, journal entry input, task management, and system shutdown—without manual intervention. Below is the step-by-step breakdown of how the system is implemented.

## 1. Setting up the XAMPP Server and React Application

The first task in the automation process is to set up the XAMPP server and start the React application. This is achieved by automating the following steps:

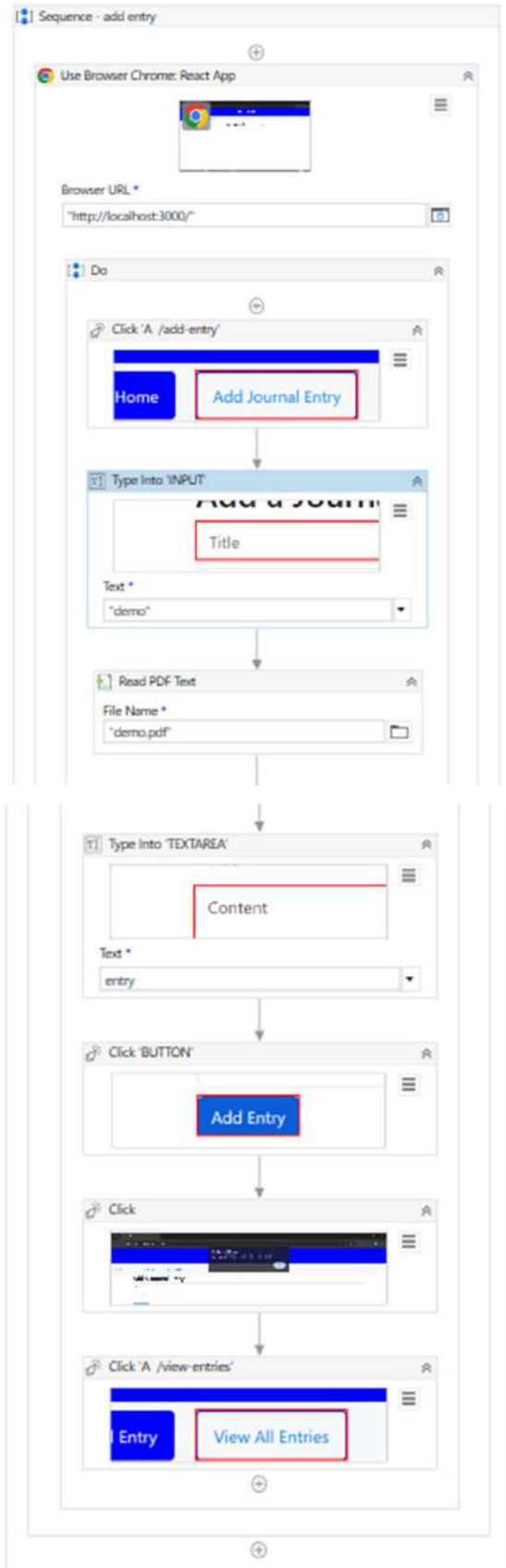
- **Launching XAMPP Control Panel:** Using UiPath's **Open Application** activity, the XAMPP Control Panel is launched. The automation then simulates clicking on the **Start** buttons for the **Apache** and **MySQL** services using **Click** activity.
- **Starting the React Application:** A command prompt is opened using the **Start Process** activity. UiPath types in the necessary command to navigate to the React project directory and run the React app using the **Type Into** activity. This ensures that the React-based journal application is launched and accessible in the browser.



## 2. Automating Journal Entry Creation from PDFs

Once the system is up and running, the next step is to automate the journal entry creation from PDFs. This is done in the following steps:

- **Reading PDF Content:** UiPath's **Read PDF Text** activity is used to extract the textual content from PDF files that contain the journal entry details. If the PDFs are scanned images, **Read PDF with OCR** can be used to extract text via optical character recognition.
- **Data Processing:** The extracted text is then processed and formatted using **String Manipulation** activities (such as **Substring**, **Replace**, **Split**, etc.) to organize the journal entry information (e.g., title, content, date) into a structured format.
- **Entering Data into the React Journal:** Using UiPath's web automation features, such as **Open Browser**, **Type Into**, and **Click**, the system enters the journal entry data into the appropriate fields of the React application. The automation can also ensure the correct fields are filled by identifying elements with the **Find Element** or **Click** activities.

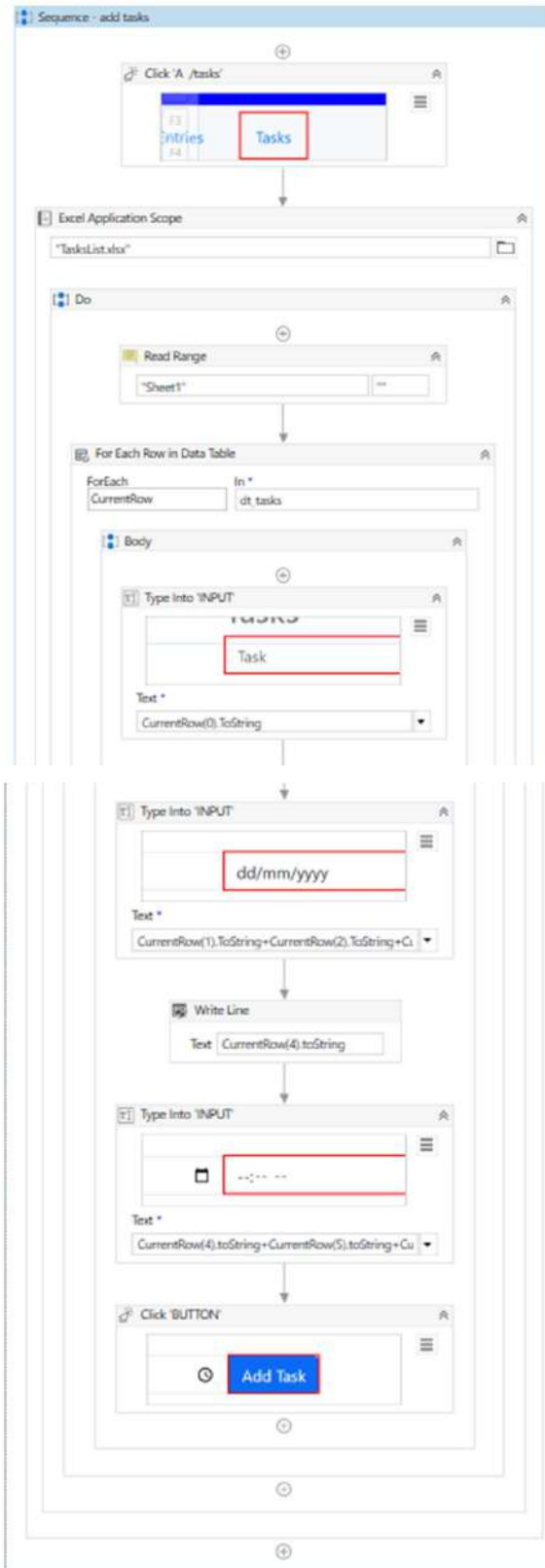


### 3. Automating Task Management from Excel

The next component is automating the task management process. Task details are stored in an Excel file and need to be added to the journal system.

- **Extracting Task Data:** UiPath's **Read Range** activity is used to extract data from the Excel file, including task names, priorities, and due dates. Each task is processed individually by iterating over the rows in the Excel sheet using a **For Each Row** loop.
- **Formatting and Adding Tasks:** Each extracted task detail is formatted as needed (using string manipulation) and then added to the journal application via the **Type Into** activity or other web automation tools. For tasks that require user selection (e.g., priority level), the **Select Item** activity is used to select the appropriate dropdown option.

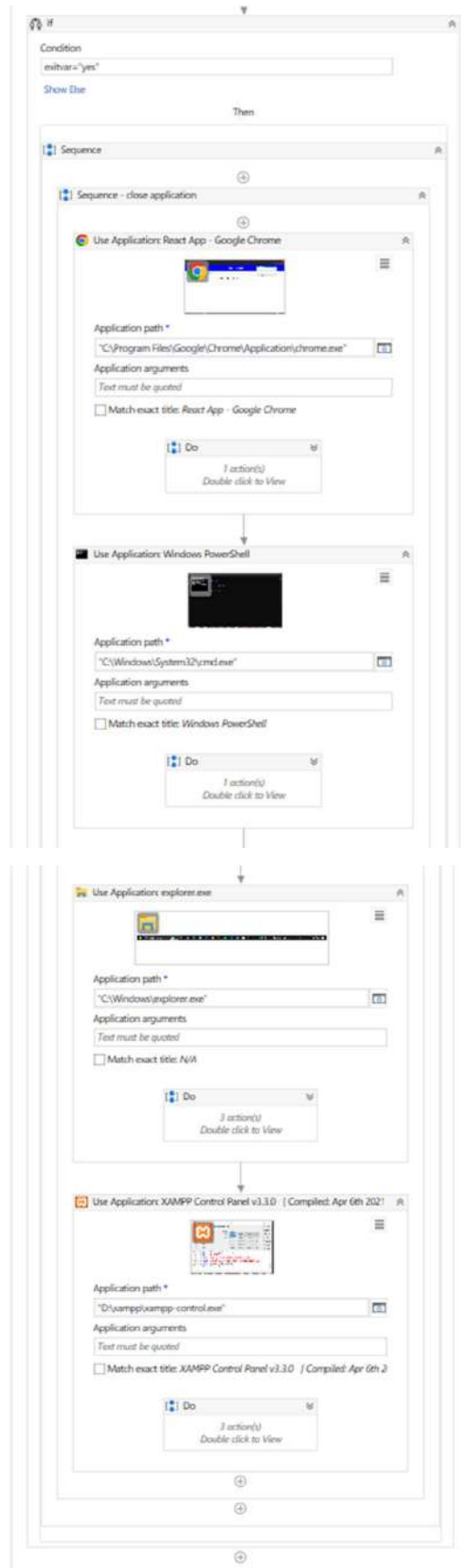




#### 4. User-Controlled Shutdown

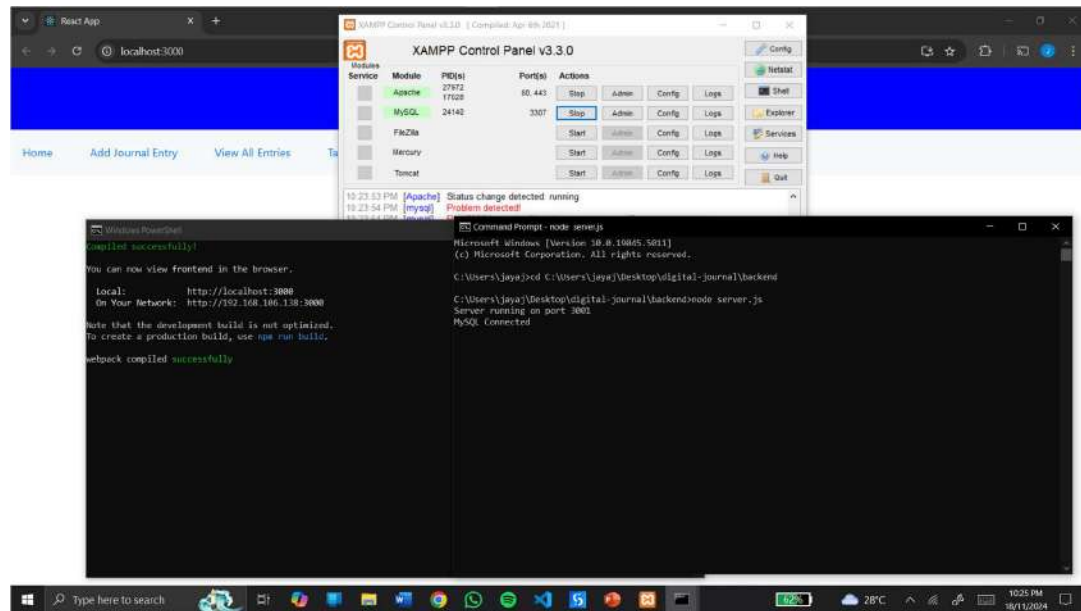
At the end of the automation process, the system prompts the user whether they want to close the applications. This is done by:

- **Prompting the User:** The **Message Box** activity in UiPath asks the user if they would like to shut down the applications. The prompt includes options like "Yes" or "No."
- **Shutting Down the System:** If the user chooses "Yes," the system will stop the XAMPP server by using the **Click** activity to interact with the XAMPP Control Panel and click the **Stop** button for Apache and MySQL. Additionally, all command prompt windows are closed using the **Close Application** activity, ensuring a clean shutdown of all related processes.

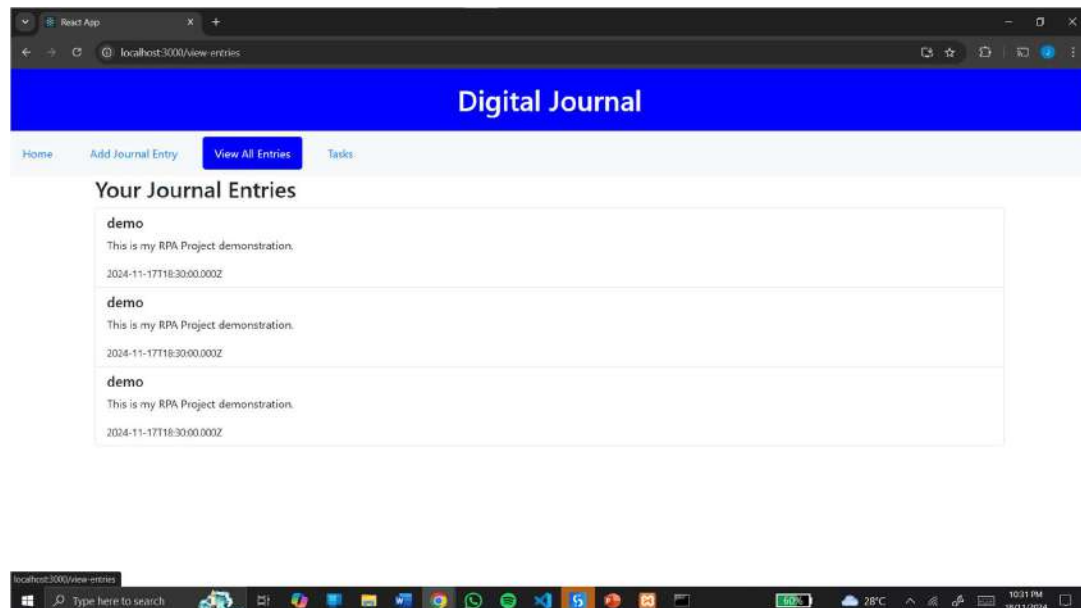


# TESTING

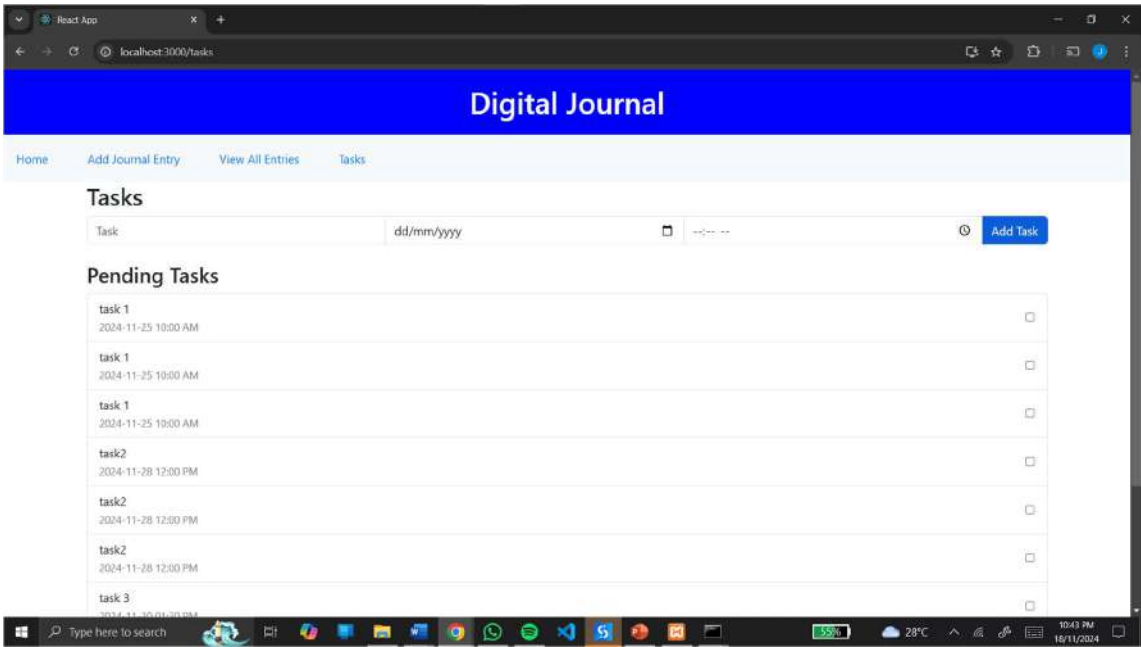
## 1. Setting up the XAMPP Server and React Application



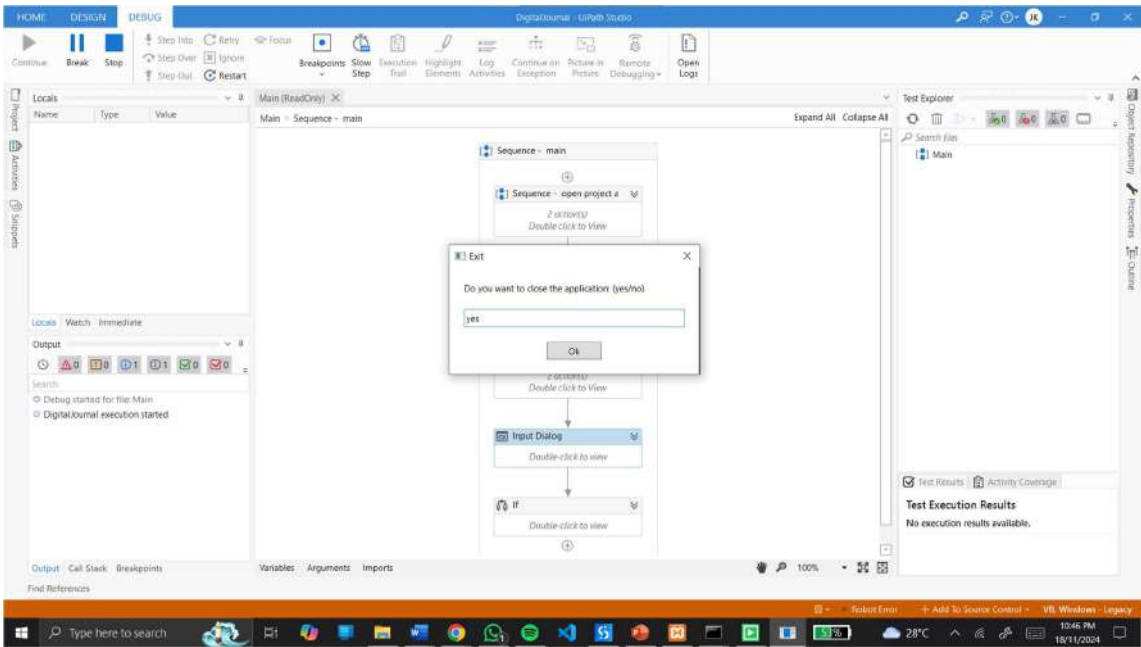
## 2. Automating Journal Entry Creation from PDFs



### 3. Automating Task Management from Excel



### 4. User-Controlled Shutdown



## CONCLUSION

The proposed automation system for managing a digital journal through Robotic Process Automation (RPA) using UiPath has been successfully developed and tested. The system automates key tasks such as starting the XAMPP server, launching the React-based journal application, extracting journal entries from PDFs, managing tasks from Excel, and shutting down the system at the user's request. By automating these tasks, the system significantly reduces manual effort, ensures greater accuracy, and improves efficiency.

The implementation demonstrates the potential of RPA in automating repetitive, time-consuming tasks in web applications. The system not only handles complex workflows with minimal human intervention but also offers a user-friendly experience with robust error handling and validation. Testing confirmed the system's functionality across different modules, including server setup, journal entry automation, and task management.

The automation is highly scalable, allowing for easy integration of additional features such as generating reports or managing larger datasets. The user-controlled shutdown feature ensures that resources are properly released after use, making the process clean and efficient.

In conclusion, the automation of the digital journal workflow through RPA not only saves time but also enhances the accuracy and consistency of tasks, making it a valuable solution for both personal and professional use in similar applications.

# **FUTURE ENHANCEMENTS**

## **1. Multi-User Support**

A key enhancement would be adding multi-user support, allowing multiple individuals to maintain their own separate journals and task lists. This can be achieved by implementing a login system where each user has their unique credentials, ensuring data privacy and personalized user experiences. The system would need to manage user accounts, and allow users to create, view, and modify their journal entries and tasks independently of others.

## **2. Cloud Backup**

Integrating cloud backup functionality would allow users to save their journal entries and tasks on a secure cloud platform, such as Google Drive, AWS, or Firebase. This would ensure that data is not lost in case of device failure and provide easy access to the journal from multiple devices, such as smartphones, tablets, or laptops. Cloud backup would also improve data synchronization, enabling seamless transitions between different devices.

## **3. Task Reminders**

Adding a reminder feature would help users stay on track with their tasks. The system would send notifications or reminders to the user about upcoming tasks and deadlines. This could be implemented using email notifications or in-app reminders, ensuring that users are always aware of important dates and deadlines. Integrating with calendar applications, like Google Calendar, could further enhance the reminder functionality, allowing users to manage tasks across platforms.

## **REFERENCES**

1. UiPath Documentation: <https://docs.uipath.com>
2. Excel Automation Tutorials, Online Resources.
3. [www.Chatgpt.com](http://www.Chatgpt.com)