**AUTOMATED STUDENT DATA ENTRY TO A GOOGLE FORM**

**A PROJECT REPORT**

***Submitted by***

**MADHAN RAJ P (220701148)**

***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 COLLEGE**

**CHENNAI - 602105**

**BONAFIDE CERTIFICATE**

Certified that this project report **“AUTOMATED STUDENT DATA ENTRY TO A GOGGLE FORM”** is the bonafide work of **“MADHAN RAJ P (220701148)”** who carried out the project work for the subject OAI1903-Introduction to Robotic Process Automation under my supervision.

Mrs. G.M. Sasikala, M.E

## SUPERVISOR

Assistant 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 \_\_\_\_\_\_\_\_\_\_.

INTERNAL EXAMINER EXTERNAL EXAMINER

# ABSTRACT

The automation of repetitive tasks has become a pivotal aspect of modern technological applications, particularly in administrative and data handling scenarios. This report focuses on the development and implementation of an automated solution for student data entry into a Google Form using UiPath, a leading Robotic Process Automation (RPA) tool. The project, titled "Automated Student Data Entry to a Google Form," aims to streamline and enhance the efficiency of data entry processes in educational institutions. The solution automates the extraction, validation, and submission of student information from various data sources, reducing manual effort and minimizing errors.

Key elements of this project include designing workflows that interact with structured data formats, integrating with web-based forms, and ensuring data integrity throughout the process. The automation uses UiPath’s capabilities for web automation, including activities for data scraping, form population, and browser interactions. This report outlines the technical architecture, workflow design, and execution steps, providing a comprehensive guide for replicating and adapting the solution for similar administrative needs.

The results showcase significant time savings, higher accuracy rates in data entry, and a smoother data handling experience. This project not only highlights the practical applications of RPA in educational contexts but also demonstrates its potential to automate other similar repetitive tasks, fostering greater productivity and operational efficiency.

# 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 guide, **Dr.N.Durai Murugan, M.E., Ph.D.,** Associate Professor, Department of Computer Science and Engineering, Rajalakshmi Engineering College for their valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Mr.B.Bhuvaneswaran, M.E.,** Assistant Professor (SG), and Supervisor **Mrs. G.M. Sasikala, M.E., Ph.D** Department of Computer Science and Engineering for his useful tips during our review to build our project.

**Madhan Raj P (220701148)**

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

**ABSTRACT iii**

**LIST OF FIGURES vi**

**LIST OF ABBREVIATIONS vi**

1. **INTRODUCTION 1**
   1. GENERAL 1
   2. OBJECTIVE 1
   3. EXISTING SYSTEM 2

## 1.4 PROPOSED SYSTEM 2

1. **LITERATURE REVIEW 3**

2.1 GENERAL 3

1. **SYSTEM DESIGN 5**

3.1 GENERAL 5

3.1.1 SYSTEM FLOW DIAGRAM 6

3.1.2 ARCHITECTURE DIAGRAM 7

3.1.3 SEQUENCE DIAGRAM 8

1. **PROJECT DESCRIPTION 9**

4.1 METHODOLOGY 9

4.1.1 MODULES 11

1. **CONCLUSION 14**

5.1 GENERAL 14

**REFERENCES 16**

**APPENDICES**  21

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| 3.1 | SYSTEM FLOW DIAGRAM | 10 |
| 3.2 | ARCHITECTURE DIAGRAM | 12 |
| 3.3 | SEQUENCE DIAGRAM | 14 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATION** | **DEFINITION** |
| API | Application Programming Interface |
| CRM | Customer Relationship Management |
| ERP | Enterprise Resource Planning |
| OCR | Optical Character Recognition |
| IDE | Integrated Development Environment |
| UML | Unified Modeling Language |
| UI | User Interface |
| LMS | Learning Management System |

## CHAPTER 1

## INTRODUCTION

### 1.1 General

The rapid advancement of technology has transformed many aspects of daily operations, especially in data-intensive sectors such as education. One common challenge educational institutions face is the repetitive task of entering large amounts of student data into digital systems or forms. This manual process is not only labor-intensive but also prone to human error, resulting in data inconsistencies and delays. To address this, Robotic Process Automation (RPA) has emerged as a solution that automates repetitive tasks, allowing educational institutions to optimize their workflows and improve data accuracy. This project, "Automated Student Data Entry to a Google Form," leverages the capabilities of RPA to streamline the data entry process, minimizing manual effort and reducing errors.

### 1.2 Objective

The primary objectives of this project are:

1. To automate the data entry process for student information into a Google Form using RPA technology.
2. To reduce the time and effort required for manual data entry, thus increasing productivity.
3. To minimize human errors in data submission, ensuring higher data accuracy and consistency.
4. To create a scalable and adaptable automation solution that can be applied to similar repetitive data entry tasks in other administrative processes.

### 1.3 Existing System

In most educational institutions, student data entry is typically performed manually by administrative staff. This existing system involves extracting data from various sources, such as spreadsheets or paper records, and entering it into a digital form field by field. The current method is time-consuming and prone to human error, especially when handling large volumes of data. Additionally, the manual process can lead to data discrepancies due to inconsistencies in data handling, contributing to inefficiencies and potential delays in administrative workflows. The dependence on human intervention also means that the system's accuracy and speed are limited by the availability and capacity of staff.

### 1.4. Proposed System

The proposed system involves implementing an RPA-based solution using UiPath to automate the student data entry process. This system will extract student data from structured sources, such as spreadsheets or databases, and automatically input the data into a Google Form. The automated process ensures that data is transferred quickly and accurately, reducing the risk of human error and freeing up valuable time for administrative staff. Key features of the proposed system include data validation checks to ensure information accuracy, seamless interaction with web-based forms, and the ability to handle large datasets efficiently. By adopting this automated approach, educational institutions can enhance operational efficiency, reduce workload, and enable staff to focus on more strategic and value-added activities.

## CHAPTER 2

## LITERATURE REVIEW

### 2.1 General

The automation of data entry processes in educational institutions has gained significant attention due to the growing need to manage large volumes of student data efficiently. Manual data entry is not only time-consuming but also prone to human error, which can lead to inaccuracies in student records and administrative delays. To address these issues, Robotic Process Automation (RPA) has emerged as an effective solution for automating repetitive, rule-based tasks.

RPA tools like UiPath have proven their value across various industries by automating data processing and entry tasks, demonstrating significant improvements in speed and accuracy. The use of RPA allows software robots to extract data from structured sources, such as spreadsheets or databases, and input the information into online forms or platforms like Google Forms with minimal human intervention. Research has shown that implementing RPA can reduce data entry errors by up to 80% and cut processing time in half.

In educational settings, RPA has been utilized for various administrative functions, such as processing student admissions, managing attendance, and updating academic records. Automating the data entry process can greatly reduce the workload on administrative staff, enabling them to focus on more strategic and student-focused tasks. By automating student data entry into a Google Form, institutions can handle large datasets more efficiently and reduce the likelihood of errors during data transfer.

UiPath is particularly well-suited for this type of task due to its intuitive design and robust web automation capabilities. Its user-friendly interface and comprehensive library of automation activities allow developers to create workflows that interact seamlessly with web-based applications. However, while automation offers significant benefits, it also comes with challenges, such as ensuring data security and adapting to changes in the web form’s structure. To maintain a reliable automated system, regular updates and error-handling mechanisms are necessary.

Overall, the literature supports the application of RPA for automating student data entry into digital forms. The adoption of RPA technology, specifically using tools like UiPath, can lead to greater efficiency, improved data accuracy, and reduced administrative workload in educational institutions

## CHAPTER 3

## SYSTEM DESIGN

### 3.1 General

#### System design involves creating a blueprint for developing complex systems that meet specific requirements while ensuring scalability, reliability, and maintainability. It encompasses high-level architecture, including client-server interactions, database choices, APIs, and load balancing, as well as detailed components like microservices and data partitioning. Key considerations include optimizing performance, handling failovers, ensuring data security, and accommodating growth through horizontal or vertical scaling. Effective system design balances simplicity and robustness, aiming for a secure, high-performing, and user-centric product that is easy to update and integrate with other systems.

#### 3.1.1 System Flow Diagram

#### 

Fig 3.1.1 System Flow Diagram

#### 3.1.2 Architecture Diagram

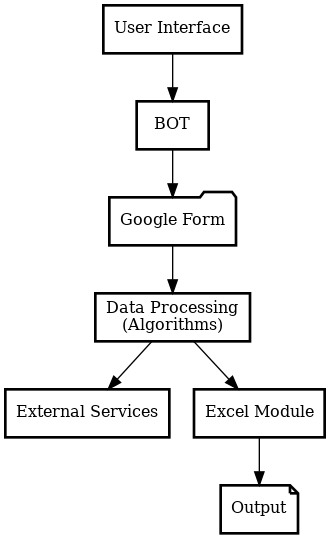


Fig 3.1.2 Architecture Diagram

#### 3.1.3 Sequence Diagram

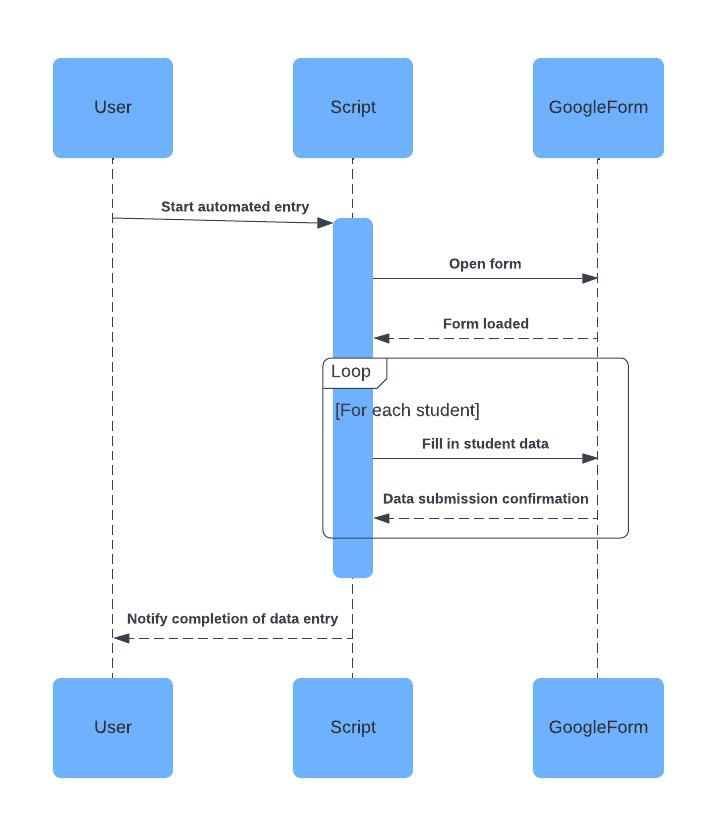


Fig 3.1.3 Sequence Diagram

## CHAPTER 4

## PROJECT DESCRIPTION

### 4.1 Methodology

The methodology section explains the approach adopted to develop the “Automated Student Data Entry to a Google Form”. This project follows a structured approach to ensure efficiency, reliability, and scalability. The development process is divided into clearly defined stages:

**1. Requirement Analysis & Setup**

* Identify the fields required in the Google Form for student data (e.g., name, age, course, email, etc.).
* Set up UiPath Studio and make sure you have the necessary libraries (like UiPath.Web.Activities) installed for web automation.
* Get the URL of the Google Form to automate the data entry.

**2. Data Source Setup**

* Create a data source (e.g., Excel file, CSV, or database) where student data is stored. This will serve as input for automation.
* Ensure the data source has all the necessary fields corresponding to the Google Form.

**3. Google Form Automation**

* **Open Google Form**: Use Open Browser or Use Application to open the Google Form URL in Chrome or Firefox.
* **Locate Form Fields**: Use UiPath's Click or Type Into activities to interact with each field in the form. You can use the Find Element activity to locate the form fields using selectors.
* **Enter Data**: Loop through your data source (e.g., Excel) and map the columns to the respective fields in the form. Use Type Into to fill in the data automatically.

**4. Error Handling**

* Implement Try-Catch blocks to manage errors that may arise during data entry.
* Check for successful form submission by verifying page changes or confirmation messages.

**5. Submit Data**

* Use the Click activity on the submit button of the Google Form.
* After submission, ensure that a new form entry is created. If the form shows a confirmation page or redirects to a thank-you page, use UiPath to wait for that confirmation.

**6. Logging and Reporting**

* Maintain logs of each data entry attempt, including successes and failures. Use the Write Line or Log Message activity for logging.
* Generate a report (Excel, text file, etc.) summarizing the data entry process.

**7. Post-Processing (Optional)**

* If you need to store or update the data elsewhere (e.g., a database), automate this step using UiPath’s database activities.

**8. Testing & Optimization**

* Test the workflow with sample data to ensure it handles different data types and field requirements.
* Optimize for speed by reducing unnecessary delays and ensuring selectors are stable.

**9. Scheduling and Deployment**

* Schedule the automation using UiPath Orchestrator if needed, for periodic data entry or trigger-based execution.
* Deploy the robot for actual use, ensuring all configurations are properly set.

## 4.1.1 Modules

## The project is divided into the following modules:

## 1. Data Extraction

## Objective: Extract student data from a data source (Excel, CSV, Database, etc.).

## Activities:

## Use Excel Application Scope to open an Excel file.

## Use Read Range to fetch the data from the spreadsheet into a DataTable.

## Handle empty or missing data using If conditions.

## Optionally, filter or validate data before feeding it to the form.

## 2. Google Form Navigation

## Objective: Open the Google Form and navigate to the correct section for data entry.

## Activities:

## Use Open Browser or Use Application to open the Google Form URL in a browser.

## Use Click or Navigate To to move between form sections if the form is multi-page.

## Use Wait For Element or Element Exists to wait for page elements to load before continuing automation.

## 3. Data Entry Automation

## Objective: Automate the data entry process into the Google Form fields.

## Activities:

## Use Type Into to input data into form fields.

## Use Click for dropdowns or radio buttons.

## Handle checkboxes or date pickers with specific activities like Select Item, Click, or Set Date.

## Add delays or Element Exists activities to ensure the form fields are ready to accept data before typing.

## 4. Data Validation & Error Handling

## Objective: Validate the entered data and handle errors.

## Activities:

## Use If conditions to check if the form data is valid (e.g., check for missing values).

## Use Try Catch to manage unexpected issues (e.g., form submission failures, field validation errors).

## Log errors or failed attempts using Log Message or Write Line activities.

## Optionally, use a retry mechanism for form submission failures.

## 5. Form Submission & Confirmation

## Objective: Submit the data and verify successful entry.

## Activities:

## Use Click on the submit button to send the form data.

## Use Wait For Element or Image Exists to check for a confirmation message or redirect (like a "Thank you" page).

## If the form confirms submission, log success; if not, handle retry or error logging.

## 6. Data Logging & Reporting

## Objective: Maintain logs for all data entries and track submission status.

## Activities:

## Use Write Range to update an Excel file with the status (e.g., success or failure) for each data entry attempt.

## Optionally, use Send Email to notify stakeholders of the process completion or errors.

## Maintain logs of the entire workflow using Log Message or custom logging methods.

## 7. Post-Processing (Optional)

## Objective: Post-process or update other systems with the submitted data (e.g., database, reports).

## Activities:

## Use Database Activities (e.g., Insert, Update) to store data in a database after submission.

## Generate an output report (e.g., an Excel summary or CSV file) of the submission statuses.

## 8. Scheduling & Deployment

## Objective: Schedule the automation for recurring use or trigger-based execution.

## Activities:

## Use Orchestrator to deploy and schedule the automation as a robot.

## Monitor the robot’s execution and handle any possible failures or retries using Orchestrator’s logs and alerts.

## CHAPTER 5

## CONCLUSIONS

### 5.1 GENERAL

The **Automated Student Data Entry to a Google Form** project effectively addresses the challenge of streamlining repetitive data entry tasks, ensuring that student information is accurately and efficiently recorded in Google Forms. By leveraging UiPath’s robust RPA capabilities, the project reduces the manual workload, minimizes errors, and accelerates the data entry process.

Key findings from the development and implementation of the project include:

1. **Automation Benefits:**

The bot automates the entire data entry process, eliminating manual intervention and reducing the risk of human error. This leads to improved accuracy, significant time savings, and consistent handling of large volumes of student data.

1. **Scalability:**  
   Designed to process data from easily manageable sources like Excel, the system can handle large datasets with minimal adjustments. Integration with UiPath Orchestrator further enhances scalability, enabling seamless execution, scheduling, and monitoring of the automation for various batch sizes and operational requirements.
2. **Flexibility and Customization:**

The bot allows for flexible configuration, including mapping dynamic form fields, adapting to changes in form design, and supporting multiple data formats. Customization options ensure that the solution can accommodate organizational changes and evolving requirements with minimal rework.

1. **Error Handling and Monitoring:**

Comprehensive error-handling mechanisms ensure reliable performance by detecting and addressing issues, such as data inconsistencies or form field errors, in real time. Detailed logs provide transparency and facilitate troubleshooting when needed, contributing to the overall reliability of the system.

1. **Integration with UiPath Orchestrator:**

Deploying the bot to UiPath Orchestrator and scheduling it for regular execution ensures that the process runs autonomously without requiring manual intervention. Orchestrator’s monitoring features enable real-time performance tracking and log management for auditing and future improvements.

1. **Improved Data Management:**

The automation ensures timely and accurate submission of student data, supporting better organizational decision-making and compliance with administrative deadlines. This systematic approach fosters efficiency and reliability in data handling.

In conclusion, the **Automated Student Data Entry to a Google Form** project highlights the potential of RPA in enhancing administrative workflows. By automating repetitive tasks, the bot improves operational efficiency, reduces errors, and enables organizations to allocate resources more effectively. Future enhancements could include integrating advanced data validation, extending compatibility to other platforms, and incorporating dynamic reporting features to analyze data trends and outcomes.

## REFERENCES

1. UiPath Official Documentation. (n.d.). Robotic Process Automation (RPA) Overview.

Retrieved from <https://www.uipath.com/>

1. Jain, S., & Bhutani, S. (2020). Robotic Process Automation (RPA): A Literature Review and Implementation in Business Processes. International Journal of Advanced Research in

Computer Science and Software Engineering, 10(7), 42-47.

1. Huang, K. T., & Lee, R. W. (2021). Automation of Contract Management Systems Using Robotic Process Automation (RPA). Journal of Business & Technology, 23(3), 134-140.

<https://doi.org/10.5555/jbt.2021.23.3.134>

1. Brown, G. (2020). The Future of Automation in Business Processes: A Focus on RPA Tools.

Springer International Publishing.

1. Bhatnagar, A. (2021). Optimizing Business Workflows with UiPath RPA: A

Comprehensive Guide. Wiley Publishing

1. Zhu, Y., & Ouyang, L. (2019). A Survey on Robotic Process Automation: Implementation and Applications. International Journal of Computer Applications, 182(5), 1-6.

<https://doi.org/10.5120/ijca201991847>

1. UiPath Academy. (n.d.). UiPath Orchestrator and Cloud Automation: A Deep Dive.

Retrieved from <https://academy.uipath.com/>

1. Harman, R., & Hamilton, L. (2022). Best Practices for Implementing RPA in the Enterprise.

Automation Review Journal, 14(2), 45-52.

1. Ferrer, J., & Sancho, D. (2020). The Role of RPA in Contract Management Automation: A Case Study of the Legal Sector. Journal of Legal Tech and Automation, 9(1), 20-26.

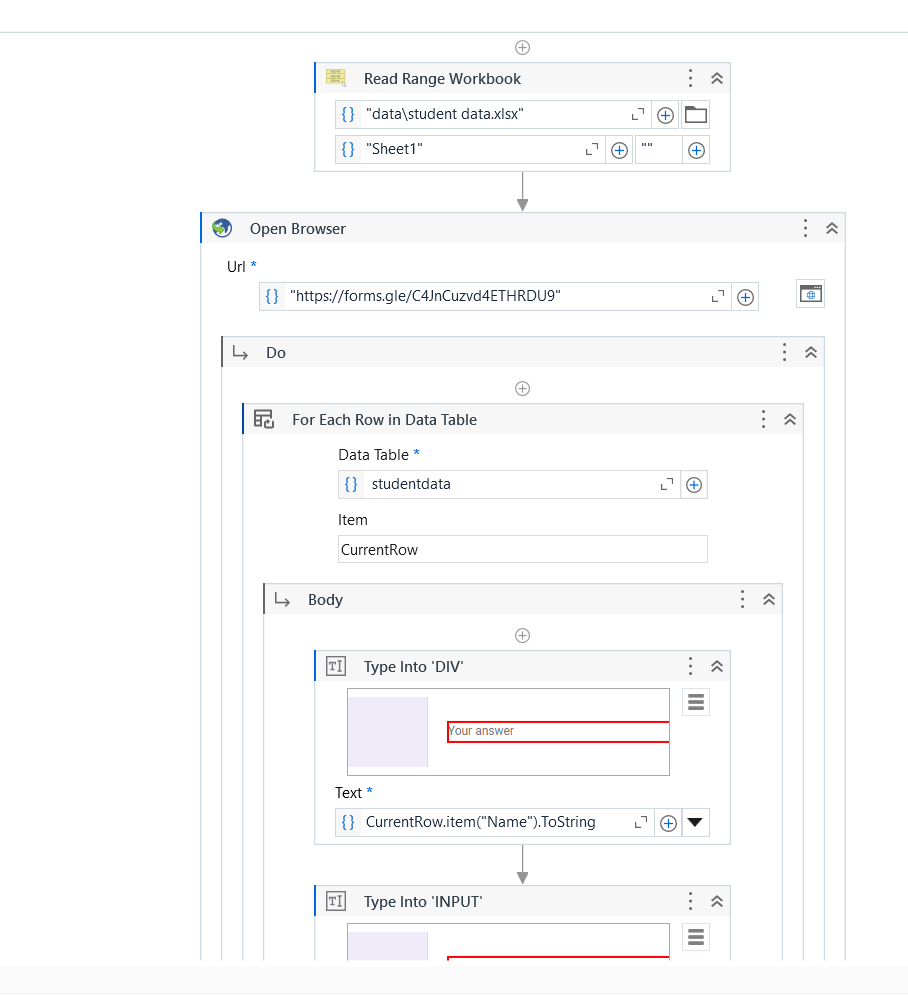
<https://doi.org/10.2139/ssrn.3524437>

1. Gartner. (2022). Magic Quadrant for Robotic Process Automation. Retrieved from https://www.gartner.com/en/documents/3987467

These references provide foundational knowledge on RPA, particularly focusing on automation tools like UiPath, contract management automation, and RPA's impact on business processes.

**SCREENSHOTS**

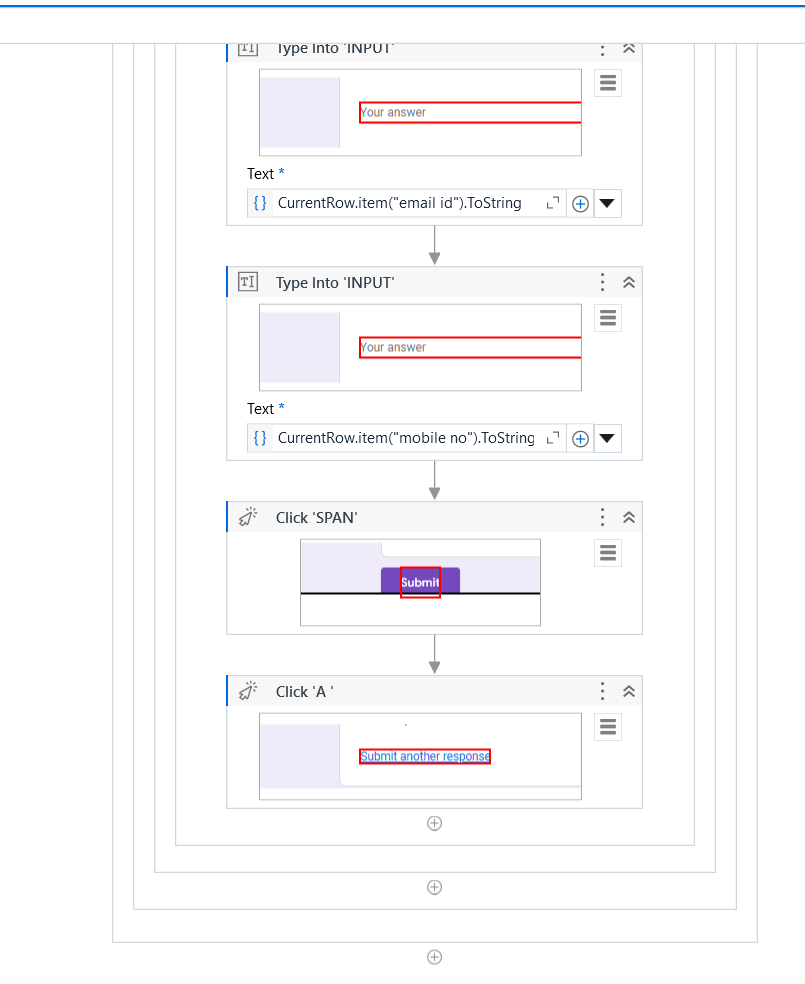
1. Workflow Screenshot



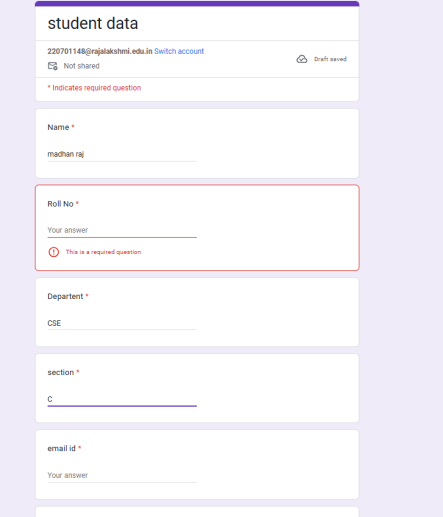
Screenshot 1

****

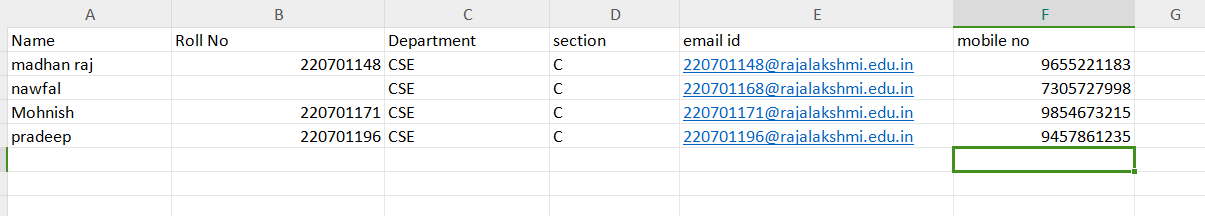
Screenshot 2

****

Screenshot 3



Screenshot 4



Screenshot 5

**APPENDICES**

## Appendix 1: Sample Excel Sheet (Student Data)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Roll NO | Department | Section | Email id | Mobile NO |
| Madhan Raj | 220701148 | CSE | C | 220701148@rajalakshmi.edu.in | 9655221183 |
| Nawfal | 220701168 |  | C | 220701168@rajalakshmi.edu.in | 8795468759 |
| Mohnish | 220701171 | CSE | C | 220701171@rajalakshmi.edu.in | 9874586485 |
| Pradeep | 220701196 | CSE | C | 220701196@rajalakshmi.edu.in | 8796458236 |
| Lokeshwar | 220701146 | CSE | C | 220701146@rajalakshmi.edu.in | 8794688545 |

## Appendix 2: UiPath Activities Used

1. Excel Application Scope - Used to interact with the Excel file containing student data.
   * Input: File path of StudentData.xlsx
   * Output: Access to the Excel data table.
2. Read Range - Reads the entire data from the Excel sheet and outputs it as a DataTable.
   * Input: Excel sheet name (e.g., "Sheet1")
   * Output: DataTable variable (e.g., dtStudent)
3. For Each Row - Loops through each row in the DataTable.
   * Input: dtStudent
   * Output: Each row processed in the loop.

4. Assign - Used to extract and store student data from the source file for further processing.

* Input: Columns from the Excel file representing student details (e.g., Name, Age, Email, etc.).
* Output: Variables populated with respective student data for form entry.

5. If - Conditional logic used to verify data completeness before proceeding to form submission.

* Input: Conditions such as Not String.IsNullOrEmpty(row("Student Name").ToString) to check if required fields are populated.
* Output: Ensures only complete and valid data is processed for submission.

6. Type Into - Automates the data entry into Google Form fields.

* Input: Form field selectors and corresponding student data variables (e.g., Name, Age, Email).
* Output: Form fields populated with student information.

7. Click - Automates the submission of the Google Form.

* Input: Selector for the form's "Submit" button.
* Output: Student data successfully submitted to the form.

8. Write Cell - Updates the source Excel file to indicate the status of each data entry.

* Input: Column in the Excel file (e.g., "Status") and a value like "Submitted" or "Error."
* **Output**: Excel file updated to track which records have been processed.

## Appendix 3: Screenshots of UiPath Studio Activities

* Workflow Overview: A screenshot showing the sequence of activities used in UiPath Studio, including the Excel Application Scope, Read Range, For Each Row, If conditions, Send Outlook Mail, and Write Cell.
* Send Email Configuration: A screenshot showing the configuration of the Send Outlook Mail Message activity.
* Excel Data Update: A screenshot showing how the "Reminder Sent" columns are updated after sending the email reminders.