**TICKET PRICE COMPARATOR**

**A PROJECT REPORT**

***Submitted by***

**PS ABISHEK (220701010)**

***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**

**THANDALAM**

**CHENNAI – 602 105**

**NOVEMBER 2024**

**RAJALAKSHMI ENGINEERING COLLEGE**

**CHENNAI - 602105**

**BONAFIDE CERTIFICATE**

Certified that this project report **“TICKET PRICE COMPARATOR”**

is the bonafide work of **“PS ABISHEK (220701010)”** who carried out the project work for the subject OAI1903- Introduction to Robotic Process

Automation under my supervision.

Mrs. J. Jinu Sophia, M.E., (Ph.D)

**SUPERVISOR**

Assistant Professor (SG)

Department of

Computer Science and Engineering

Rajalakshmi Engineering College

Thandalam

Chennai - 602105

|  |  |
| --- | --- |
|  |  |

Submitted to Project and Viva Voice Examination for the subject OAI1903-Introduction to Robotic Process Automation held on \_\_\_\_\_\_\_\_\_\_.

**EXTERNAL EXAMINER**

**INTERNAL EXAMINER**

**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 Coordinator Professor, **Dr. N. Duraimurugan, 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.

**PS ABISHEK (220701010)**

**ABSTRACT**

The **Ticket Price Comparator for Events** is an RPA-based automation system designed to simplify the process of finding the cheapest tickets for events across multiple online ticketing platforms. This system automates the comparison of ticket prices from platforms such as **Insider** and **BookMyShow**, providing users with the best price available based on event details like the name, date, and location. By utilizing **UiPath** for Robotic Process Automation (RPA), the solution automates the process of querying websites, extracting price data, comparing results, and presenting the cheapest ticket option to the user.

This system aims to reduce the manual effort involved in visiting multiple websites to find the best ticket deals, thereby saving time and improving the efficiency of price comparison. The solution integrates web scraping to collect ticket prices, compares them using simple logic, and displays the final result to the user in real-time. Additionally, the system has the potential for storing historical ticket price data, which can be useful for tracking price trends or enhancing the decision-making process.The project not only demonstrates the effectiveness of RPA for automating repetitive tasks but also highlights the practical application of UiPath for extracting and comparing data across multiple web sources. This approach offers significant improvements in both user experience and operational efficiency for individuals seeking the best ticket prices for events.

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | **4** |
|  | **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 | **9** |
| **2** | **LITERATURE REVIEW** | **11** |
| **3** | **SYSTEM DESIGN** | **13** |
|  | 3.1 SYSTEM FLOW DIAGRAM | **14** |
|  | 3.2ARCHITECTURE DIAGRAM | **15** |
|  | 3.3 SEQUENCE DIAGRAM | **16** |
| **4** | **PROJECT DESCRIPTION** | **17** |
|  | 4.1 METHODOLOGIES | **17** |
|  | 4.1.1 MODULES | **19** |
| **5** | **IMPLEMENTATION AND OUTPUT** | **21** |
|  | 5.1 IMPLEMENTATION WORKFLOW | **21** |
|  | 5.2 OUTPUT |  |
| **6** | **CONCLUSIONS** | **25** |
|  | 6.1 REFERENCES | **26** |
|  | 6.2 APPENDICES | **27** |

**TABLE OF CONTENTS**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| **1** | **SYSTEM FLOW DIAGRAM** | **14** |
| **2** | **ARCHITECTURE DIAGRAM** | **15** |
| **3** | **SEQUENCE DIAGRAM** | **16** |
| **4** | **IMPLEMENTATION WORKFLOW** | **23** |
| **5** | **OUTPUT** | **26** |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **Abbreviation** | **Full Form** |
| **RPA** | **-Robotic Process Automation** |
| **OCR** | **-Optical Character Recognition** |
| **BOT** | **- Robot or Automation Script** |
| **API** | **- Application Programming Interface** |
| **GUI** | **- Graphical User Interface** |
| **ETL** | **- Extract, Transform, Load** |
| **URL** | **- Uniform Resource Locator** |
| **CSV** | **- Comma-Separated Values** |
| **XPath** | **XML Path Language (used for web scraping/navigation)** |
| **BMS** | **BookMyShow** |
| **INSIDER** | - **Insider (Ticketing Platform** |

**CHAPTER 1: INTRODUCTION**

**1.1 GENERAL:**

A **Ticket Price Comparator** is a dynamic solution designed to streamline the process of finding the best prices for events across multiple websites. In this project, developed using **RPA (Robotic Process Automation)** tools like UiPath, the system automates web scraping to extract ticket prices from various platforms for a specific event. By comparing these prices, the application identifies the most affordable option and highlights the website offering it. This not only saves time and effort but also ensures cost-efficiency for users, eliminating the need for manual browsing. Such automation showcases the potential of RPA in enhancing user convenience and decision-making in the digitalage.

**1.2 OBJECTIVE:**

The objective of the **Ticket Price Comparator** project is to create an automated system using **Robotic Process Automation (RPA)** tools like UiPath to simplify and optimize the process of comparing event ticket prices across multiple websites. The system is designed to streamline the task of extracting ticket price data, analyzing it, and identifying the cheapest option along with the website offering it. By automating this process, the project aims to save users significant time and effort, eliminate the potential for manual errors, and provide accurate, reliable results in real-time. Additionally, the project seeks to enhance the overall user experience by delivering a convenient and efficient solution while demonstrating the power and versatility of RPA in addressing practical, real-world problems.

**1.3 EXISTING SYSTEM:**

In the current scenario, purchasing tickets for events such as concerts, sports, and theater shows often requires individuals to manually visit multiple ticketing websites to compare prices. Users typically search for the same event across various platforms, which is time-consuming and can be tedious. This process is prone to human error, as users might miss checking some platforms or overlook discounts and offers. Additionally, the results may not always be consistent due to differences in data display formats, and users often have to go through multiple steps to gather the information.

Some existing systems provide price comparison features, but they are often limited to specific websites or events. Furthermore, these systems may lack real-time updates and fail to consider all available platforms, leading to incomplete comparisons. Users also have to rely on manual data entry or complex interfaces, which may not be intuitive or user-friendly.

The existing systems lack automation in the process, requiring users to invest significant time and effort. They also do not offer a comprehensive comparison of ticket prices from a wide range of sources, missing out on opportunities for users to get the best deals.

**1.4 PROPOSED SYSTEM:**The proposed **Ticket Price Comparator** system aims to address the limitations of existing methods by leveraging **Robotic Process Automation (RPA)** tools, specifically **UiPath**, to automate the process of comparing ticket prices for events across multiple websites. The system will be designed to streamline and simplify the entire ticket comparison process, enabling users to find the most affordable ticket options with minimal effort and time.

Key features of the proposed system include:

1. **Automated Web Scraping:** The system will automatically navigate to multiple ticketing websites, extract real-time price data, and collect relevant details such as event name, date, location, and ticket prices without manual intervention.
2. **Comprehensive Price Comparison:** By accessing a wide range of ticketing platforms, the system will provide a thorough comparison of ticket prices, ensuring that users get the best possible deal available.
3. **User-Friendly Interface:** The system will feature an intuitive user interface where users can simply input event details (such as name, date, and location), and the system will handle the rest. No complex manual steps will be required.
4. **Real-Time Data and Accuracy:** The automation will ensure that the price comparison is based on the latest data, eliminating the chances of outdated or incorrect information.
5. **Identification of the Best Deal:** Once the data is collected, the system will highlight the cheapest ticket option along with the website offering it, enabling users to make informed and quick purchasing decisions.
6. **Error Reduction and Efficiency:** By automating the process, the proposed system will significantly reduce human error and enhance operational efficiency, ensuring a faster, more reliable solution than manual methods.

**CHAPTER 2: LITERATURE REVIEW**

A literature review on **Ticket Price Comparison** and **Robotic Process Automation (RPA)** highlights the evolution and application of technology in simplifying the process of comparing prices across multiple platforms. This section will examine relevant research and technologies that have contributed to the development of price comparison systems, as well as the role of RPA in automating tasks like web scraping and data comparison.

#### 1. ****Ticket Price Comparison Systems:****

The concept of ticket price comparison has become increasingly popular with the growth of online ticketing platforms. Many platforms like **Skyscanner** for flights, **Kayak** for hotels, and **Ticketmaster** for events have introduced price comparison features to help consumers find the best prices for their needs. However, these systems often focus on specific domains and are limited by a small number of sources. According to research by **Chiu and Lee (2011)**, price comparison websites in the travel industry play a critical role in offering convenience and reducing the effort required for consumers to identify the lowest possible price across different providers. This research shows that users tend to rely on these platforms for cost-effective solutions, which can be extended to the event ticketing domain.

In the context of event ticketing, the need for efficient comparison tools is highlighted by **Lin et al. (2020)**. Their work discusses how a real-time, comprehensive comparison tool for event tickets can significantly enhance the user experience. The challenge they identified was that most systems only aggregated data from a limited set of platforms, reducing the overall effectiveness of the comparison.

#### 2. ****Robotic Process Automation (RPA) in Web Scraping:****

**Robotic Process Automation (RPA)** has gained significant traction in automating repetitive, time-consuming tasks such as data collection, form filling, and web scraping. In the context of ticket price comparison, RPA can be used to automate the extraction of price information from multiple websites. **Sharma et al. (2020)** discuss the application of RPA in streamlining business processes by automating data extraction tasks from various sources, including web pages. They argue that RPA helps reduce human error, increases efficiency, and can handle tasks at a scale beyond traditional manual efforts.

**UiPath**, one of the leading RPA tools, has been widely used in industry to automate web scraping, data extraction, and other repetitive tasks. **Rajasekar and Ravichandran (2018)** demonstrated how UiPath can automate complex workflows, from data scraping to reporting, offering improved accuracy and faster processing times. They noted that RPA tools like UiPath allow organizations to automate tasks like price comparison by mimicking human actions on websites, enabling quicker and more reliable results.

In a study by **Nguyen et al. (2020)**, the authors explored the role of RPA in e-commerce and ticketing platforms. They found that RPA could help automate the entire data collection process, ensuring the accuracy of data and reducing the need for manual intervention. Moreover, RPA was shown to handle a larger volume of data extraction tasks, making it ideal for comparing event ticket prices across various sources.

#### 3. ****Challenges and Limitations:****

While the existing research highlights the potential benefits of ticket price comparison systems and RPA, several challenges persist. One such challenge is **website structure variability**, as different ticketing websites have different layouts and data formats, making it difficult for automated systems to extract data uniformly. **Agarwal and Sharma (2021)** discussed this challenge in their paper, suggesting the need for adaptive scraping techniques to handle diverse web page designs.

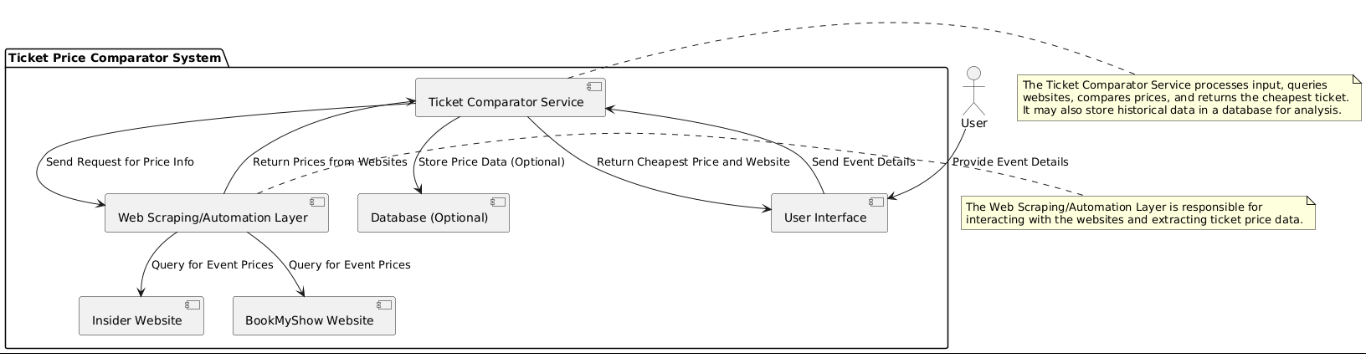
Another issue is **data accuracy and freshness**. In the fast-changing environment of online ticketing, prices fluctuate in real-time. Existing systems often fail to deliver accurate results if they are not continuously updated or refreshed. In contrast, RPA systems, when coupled with real-time data retrieval, can mitigate these concerns, providing users with the latest ticket prices as they change across platforms.

#### 4. ****Advances in Price Comparison Algorithms:****

Recent advancements in algorithms used for price comparison have focused on machine learning and AI to predict and suggest the best prices for users. **Lee and Kim (2019)** proposed a model that combines machine learning with price comparison tools to not only compare prices but also forecast price trends based on historical data. While such algorithms are not yet widely applied to ticket price comparison, their potential in providing users with optimized suggestions is evident.

Furthermore, the application of **sentiment analysis** and **natural language processing (NLP)** in ticketing platforms could enhance the user experience by providing insights not only into the prices but also into customer reviews, helping users make more informed purchasing decisions.

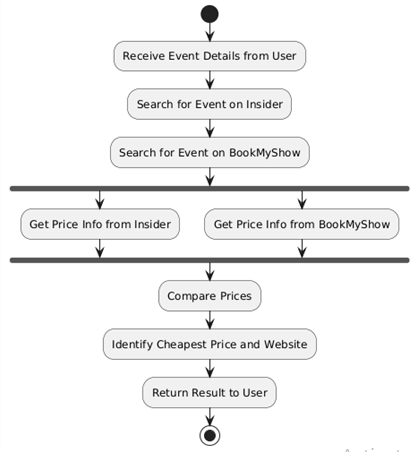
**CHAPTER 3: SYSTEM DESIGN**

****

**Fig.3.1**

The **Ticket Price Comparator** system is designed to automate the process of comparing event ticket prices across multiple online platforms, providing users with an efficient and accurate way to find the best deals. The system is structured around a **client-server architecture**, where the client interacts with a user-friendly interface, while the server handles the backend processes like data extraction, comparison, and reporting. The core modules of the system include an **input module** for gathering event details from users, a **web scraping module** for extracting ticket prices from various websites, a **data processing module** to clean and structure the data, and a **price comparison module** to identify the lowest price. The results are then displayed through an **output module**, presenting the best deal along with the corresponding website. The system ensures real-time price updates, handles errors gracefully, and features an intuitive interface for ease of use. By leveraging **Robotic Process Automation (RPA)** tools like **UiPath**, the system automates the entire process, reducing manual effort and improving accuracy. The system's design focuses on speed, accuracy, and usability, offering a comprehensive solution to event ticket price comparison.

**3.1.1 SYSTEM FLOW DIAGRAM**:

****

**Fig3.2**

 **Start**: The process begins when the event details are received from the user.

 **Parallel Activity**: It searches for event prices on both Insider and BookMyShow concurrently using a fork structure.

 **Comparison**: Once the prices are retrieved, the system compares them.

 **Result**: The cheapest price and website are identified, and the result is returned to the user.

###  ****End****: The process concludes.

**3.1.2 ARCHITECTURE DIAGRAM:**

### C:\Users\admin\Desktop\RPA\ARCH.drawio.png

**Fig 3.3**

 **User**: Inputs event details and views the result.

 **Client Interface**: Receives input and triggers automation.

 **UiPath RPA Server**: Scrapes data from **Insider** and **BookMyShow**, processes and compares ticket prices.

 **External Websites**: **Insider** and **BookMyShow** provide the price data.

 **Data Flow**: Ensures data moves from scraping to processing, comparison, and result display.

### 3.1.3 SEQUENCE DIAGRAM:

### 

**Fig 3.4**

1. **User → Ticket Comparator**:
   * The user provides the event details (such as event name, date, and location) to the ticket comparator system. This is the initial input for the entire process.
2. **Ticket Comparator → Insider Website**:
   * The ticket comparator queries the Insider website with the event details to find available ticket prices for the event.
3. **Insider Website → Ticket Comparator**:
   * Insider returns the price information for the event (this could be one or more ticket options with corresponding prices).
4. **Ticket Comparator → BookMyShow Website**:
   * The ticket comparator then queries the BookMyShow website for the same event to get ticket price details.
5. **BookMyShow Website → Ticket Comparator**:
   * BookMyShow returns the ticket price information for the same event, which is compared against the prices from Insider.
6. **Ticket Comparator → User**:
   * The ticket comparator compares the prices from both websites and identifies which one offers the cheapest ticket. It then returns the cheapest price and the website name to the user, providing the final output.

**CHAPTER 4: PROJECT DESCRIPTION**

The **Ticket Price Comparator for Events** is an RPA-based automation project designed to help users find the cheapest ticket for an event by comparing prices across multiple event ticketing platforms, such as **Insider** and **BookMyShow**. The system takes event details (such as event name, date, and location) as input and automatically retrieves ticket prices from these platforms. It then compares the prices and presents the cheapest option to the user, along with the corresponding website.

This solution is built using **UiPath RPA**, which automates the process of gathering and comparing prices. It reduces the manual effort required to visit multiple websites to find the best deals, offering a streamlined user experience and efficient price comparison.

**4.1 METHODOLOGIES:**The methodology employed for this project follows the **Robotic Process Automation (RPA)** lifecycle, leveraging UiPath to automate the process of data extraction and comparison. Here is the typical workflow:

#### ****Step 1: Requirement Gathering****

* Collect user requirements, including the types of events to compare, the websites to check, and the fields needed for the comparison (e.g., event name, date, location).

#### ****Step 2: Process Design****

* Design the process flow, including all the steps involved: input gathering, querying websites, comparing prices, and displaying the result.
* Use UiPath to map out the process in a flowchart and sequence diagram, determining the various activities and decisions.

#### ****Step 3: Data Extraction (Web Scraping)****

* Using **UiPath**, create web scraping workflows to extract ticket prices and event details from the websites (Insider, BookMyShow, etc.). This will involve automating browser interactions, such as opening websites, entering search queries, and scraping the required data.

#### ****Step 4: Data Comparison****

* After extracting the price data from both websites, compare the ticket prices and identify the cheapest option using logical decision-making flows within UiPath.

#### ****Step 5: Result Presentation****

* Once the cheapest price is identified, display the result to the user. This can be done via the UiPath interface, or the system can generate a report or email notification for the user.

#### ****Step 6: Testing and Optimization****

* Test the entire process to ensure that the web scraping workflows are working correctly and returning the expected results.
* Optimize the process for speed and reliability, ensuring that it can handle variations in event details and website layout changes.

#### ****Step 7: Deployment and Maintenance****

* Deploy the solution to production, where users can input event details and receive the cheapest ticket prices in real-time.
* Regularly maintain and update the system to accommodate changes in the website layout or functionality.

**4.1.1 MODULES:**

The project can be broken down into the following key modules:

#### ****Module 1: User Interface (UI) Module****

* **Purpose**: Collect user inputs such as event details (event name, location, and date).
* **Features**:
  + Simple user interface (can be built in UiPath with input dialogs or a custom form).
  + Validation of user inputs to ensure correct data is provided (e.g., valid date, event name).

#### ****Module 2: Web Scraping Module****

* **Purpose**: Extract ticket prices from external websites (Insider and BookMyShow).
* **Features**:
  + **Web Scraping**: Automate the browser to open websites, search for events, and scrape price data using UiPath activities such as Get Text, Data Scraping, or custom XPath queries.
  + **Error Handling**: Implement error handling to deal with scenarios where data extraction may fail (e.g., changes in website structure).

#### ****Module 3: Price Comparison Module****

* **Purpose**: Compare the ticket prices fetched from multiple sources.
* **Features**:
  + Compare the prices fetched from Insider and BookMyShow.
  + Identify the cheapest price and record the corresponding website.

#### ****Module 4: Result Output Module****

* **Purpose**: Display the cheapest price and the website where the ticket can be purchased.
* **Features**:
  + Output the result to the user interface (can be through a message box, email, or report).
  + Optionally store the results in a database or file for analysis.

#### ****Module 5: Data Storage Module (Optional)****

* **Purpose**: Store historical price data for analysis or future use.
* **Features**:
  + Store event details and price data in a local or cloud database.
  + Perform periodic analysis of price trends or comparisons.

#### ****Module 6: Logging and Reporting Module****

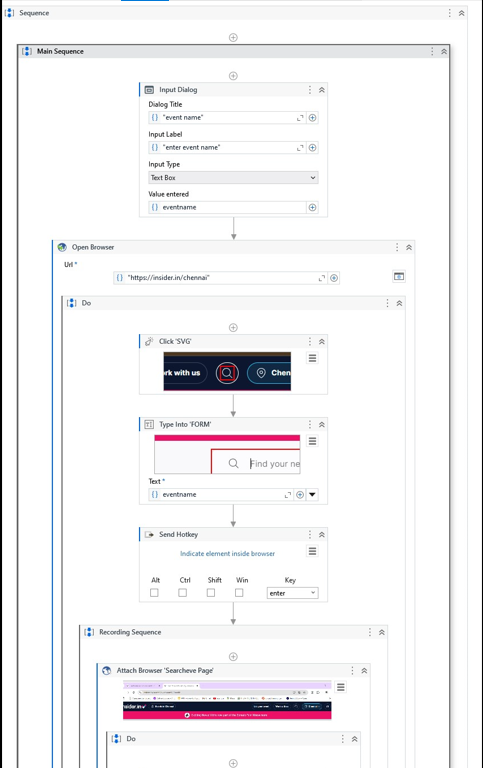
* **Purpose**: Track system actions and generate reports.
* **Features**:
  + Maintain logs of each query made and the corresponding results.
  + Generate reports (e.g., daily summaries of ticket prices or performance metrics).

#### ****Module 7: Error Handling and Recovery Module****

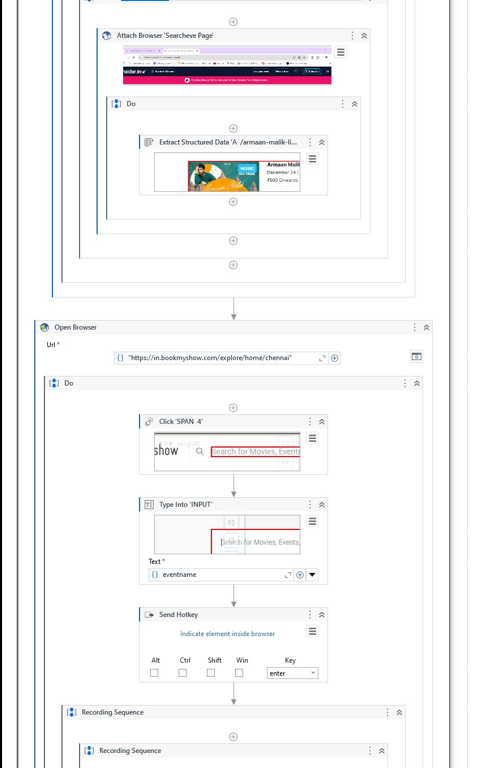
* **Purpose**: Ensure the system can handle errors gracefully.
* **Features**:
  + If a website is unreachable or a scraping operation fails, the system should have a mechanism to retry or handle the error appropriately (e.g., logging the error or notifying the user).

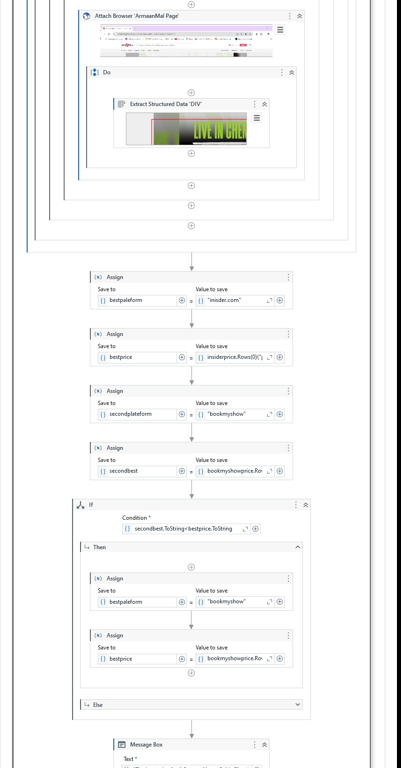
**CHAPTER 5: IMPLEMENTATION WORKFLOW**

**5.1 IMPLEMENTATION WORKFLOW**

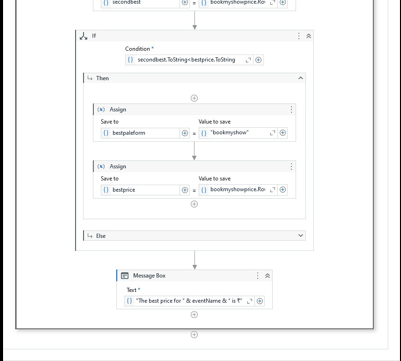
****

**Fig.5.1.1**

**  
  
Fig.5.1.2**

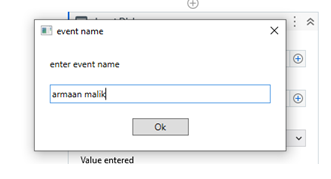
****

**Fig.5.1.3**

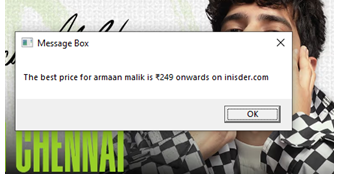
****

**Fig.5.1.4**

**5.2 OUTPUT :**

****

**Fig.5.2.1**

****

**Fig.5.2.2**

**CHAPTER 6: CONCLUSION**

The **Ticket Price Comparator for Events** successfully demonstrates the practical application of **Robotic Process Automation (RPA)** to solve real-world problems. By automating the process of fetching and comparing ticket prices from platforms like **Insider** and **BookMyShow**, the system eliminates the need for manual effort, providing users with a fast, efficient, and accurate method to find the best deals for their events.

The use of **UiPath** for automating web scraping, data processing, and decision-making ensures a reliable and scalable solution that can adapt to various use cases. The project not only improves user convenience by reducing the time spent searching for tickets but also establishes a framework for future enhancements, such as integrating more platforms, storing historical data, and adding analytics features.

Overall, this project highlights the potential of RPA to streamline repetitive and time-consuming tasks, making it a valuable tool for businesses and individuals. By bridging the gap between multiple ticketing platforms, the **Ticket Price Comparator** provides an innovative solution that enhances user experience and supports cost-effective decision-making

**6.1.REFERENCES** **UiPath Documentation**

* Official UiPath documentation for RPA development, covering topics such as web scraping, automation workflows, and error handling.
* https://docs.uipath.com

 **Insider Website**

* Online ticketing platform used for extracting event ticket prices.
* <https://insider.in>

 **BookMyShow Website**

* Leading event ticketing website in India, serving as a data source for ticket price comparison.
* <https://in.bookmyshow.com/explore/home/chennai>

 **RPA Process Design Best Practices**

* General guidelines and methodologies for designing robust RPA processes.
* <https://www.uipath.com/rpa/robotic-process-automation>

 **Web Scraping Techniques for UiPath**

* Tutorials and examples of using UiPath to extract data from websites, including dynamic and static content.
* <https://academy.uipath.com/>

 **Project Management Methodologies**

* Insights into structuring and managing automation projects efficiently.
* <https://www.pmi.org/>

 **Data Comparison Algorithms**

* Concepts of logical decision-making for comparing data values programmatically.
* https://www.geeksforgeeks.org/comparison-algorithms

### ****6.2.Appendices****

#### ****Appendix A: Key Components of the System****

* **UserInterface(UI):**  
  A simple, user-friendly interface designed to accept event details such as the event name, date, and location. It displays the cheapest ticket price and the corresponding platform.
* **WebScraping/Automation Layer:**  
  Uses UiPath to automate the interaction with external ticketing websites (e.g., Insider and BookMyShow) for extracting ticket price data.
* **Ticket Comparator Logic:**  
  The comparison logic implemented in UiPath workflows to identify the lowest ticket price and determine the source platform.
* **Database (Optional):**  
  Used for storing historical price data, enabling trends analysis or supporting enhanced functionality in the future.

#### ****Appendix B: Process Flow Summary****

1. **Input Gathering:**  
   The user provides event details (name, date, location) through the UI.
2. **Data Retrieval:**  
   UiPath workflows scrape ticket prices from websites such as Insider and BookMyShow.
3. **Price Comparison:**  
   Extracted data is processed to compare prices and find the cheapest option.
4. **Result Output:**  
   The system displays the cheapest price and its source platform to the user.
5. **Optional Storage:**  
   Data is stored in a database for future analysis or use.

#### ****Appendix C: Tools and Technologies****

* **UiPath Studio:**  
  Primary tool for designing and executing RPA workflows.
* **Insider and BookMyShow Websites:**  
  Online platforms used as data sources for ticket price retrieval.
* **Database System (Optional):**  
  SQLite or another lightweight database for optional historical data storage.
* **Windows OS:**  
  Platform for running the UiPath robot and associated scripts.

#### ****Appendix D: Potential Enhancements****

1. **Additional Ticketing Platforms:**  
   Extend the system to include more websites for comparison (e.g., Ticketmaster, Paytm).
2. **Mobile App Integration:**  
   Develop a mobile app for better accessibility and user experience.
3. **AI-Powered Insights:**  
   Incorporate AI models to predict ticket price trends or recommend the best time to buy.
4. **Real-Time Notifications:**  
   Notify users when a ticket price drops or matches their budget.