**A PRELIMENERY REPORT ON**

**FLIGHT TICKET COMPARISON BOT**

SUBMITTED TO THE VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

**BACHELOR OF TECHNOLOGY (COMPUTER ENGINEERING)**

##### SUBMITTED BY

**Group No. 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Name | PRN | Roll No. | Division |
| 1 | Darshan Vadile | 22210373 | 323072 | C |
| 2 | Atharva Karandikar | 22320134 | 323083 |



## DEPARTMENT OF COMPUTER ENGINEERING

**BRACT’S**

**VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY**

SURVEY NO. 3/4, KONDHWA (BUDRUK), PUNE – 411048, MAHARASHTRA (INDIA).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | | | | **Title of Chapter** | **Page No.** |
| **01** | | | | **Introduction** | 3 |
|  | 1.1 | | | Overview | 3 |
|  | 1.2 | | | Motivation | 3 |
|  | 1.3 | | | Problem Definition and Objectives | 3 |
|  | 1.4 | | | Project Scope & Limitations | 3 |
|  | 1.5 | | | Methodologies of Problem solving | 3 |
| **02** | | | | Literature Survey | 4 |
| **03** | | | | System Design | 5 |
|  | | 3.1 | | System Architecture | 5 |
| **04** | | | | Project Implementation | 6 |
|  | | | 4.1 | Overview of Project Modules | 6 |
|  | | | 4.2 | Tools and Technologies Used | 6 |
|  | | | 4.3 | Algorithm Details | 7 |
| **05** | | | | Results | 9 |
|  | | | 5.1 | Outcomes | 10 |
|  | | | 5.2 | Screen Shots | 10 |
|  | | | 5.3 | Metrics | 11 |
| **06** | | | | Conclusions | 12 |
|  | | 9.1 | | Conclusions | 12 |
|  | | 9.2 | | Future Work | 12 |
|  | | 9.3 | | Applications | 12 |
|  | | | | **References** | 13 |

## 1. INTRODUCTION

**1.1 Overview**

The Flight Ticket Comparison Bot is an RPA solution that automates the search for the three cheapest flights for a given source, destination, and departure date. Leveraging Google Flights, the bot extracts fare, airline, departure time, duration, and a direct booking link, presenting results to the user in a single view.

**1.2 Motivation**

Online fare‑comparison portals still require manual form‑filling and page navigation. This is tedious and error‑prone for casual users and travel desks alike. By delegating the repetitive search‑and‑compare task to an unattended bot, we cut effort, reduce human error, and obtain up‑to‑date fares instantly.

**1.3 Problem Definition & Objectives**

**Problem Statement:** Design and implement a bot that scans flight ticket prices across available online sources and returns the three cheapest options, showing: price, airline, departure/arrival time, flight duration, and booking link.

**Objectives**

1. Capture live fare data from Google Flights.
2. Sort the dataset in ascending order of price.
3. Extract the five key attributes for the top three cheapest flights.
4. Present the consolidated result to the end‑user.

**1.4 Scope & Limitations**

* *Scope*: One‑way economy searches, single departure date, data source limited to Google Flights, Automation Anywhere A360.
* *Limitations*: No support for return or multi‑city itineraries; fares may vary between scrape and booking; DOM changes on google.com/flight may require bot maintenance; currency fixed to INR.

**1.5 Methodology of Problem Solving**

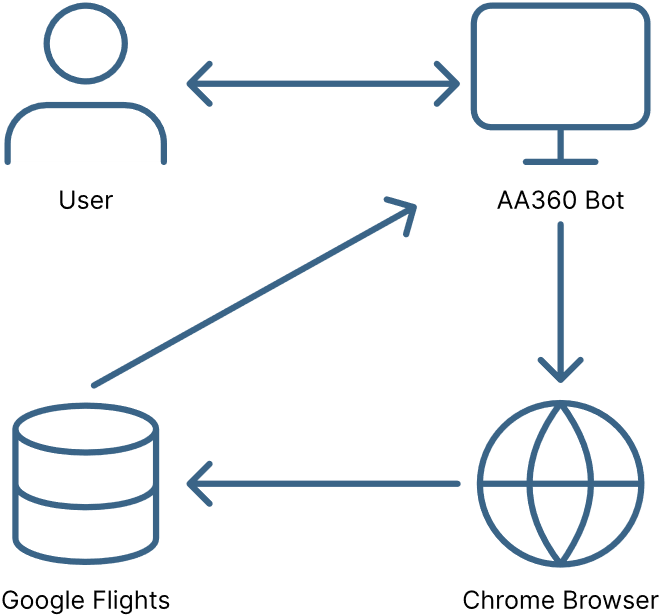
1. Accept user inputs (source IATA, destination IATA, date) through a simple form.
2. Construct Google Flights URL and launch in Chrome via Automation Anywhere.
3. After page load, use DOM XPath commands to collect flight cards into a list variable.
4. Parse each item to retrieve fare, airline, timing, duration, booking hyperlink.
5. Convert fare strings to integers, sort list, slice first three.
6. Display result set in a message box.

**2. LITERATURE SURVEY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Year** | **Author / Source** | **Approach** | **Data Source** | **Key Points** | **Limitations** |
| 2023 | Skyscanner Dev API | REST Search & Quote API | Skyscanner | Rich metadata, multi‑currency | Paid access, rate‑limited |
| 2022 | Cleartrip Public API | JSON flights endpoint | Cleartrip | India‑centric, simple auth | Limited carriers |

Gap identified: Prior work emphasises APIs; minimal literature showcases a pure RPA scraper that eliminates API costs while keeping implementation lightweight.

**3. SYSTEM DESIGN**

**3.1 System Architecture**

Components:

 **User** – Supplies three inputs (source IATA, destination IATA, departure date) and receives the ranked list of the three cheapest flights.  
 **Bot (Automation Anywhere A360)** – Orchestrates the workflow: builds the Google Flights URL, launches and controls Chrome, scrapes flight details, ranks them by price, and returns the results.  
 **Chrome Browser** – A browser instance driven by the bot; it renders the Google Flights page so the bot can interact with live elements exactly as a human would.  
 **Google Flights** – The external web service that aggregates real‑time flight schedules and fares; it is the single data source the bot queries.

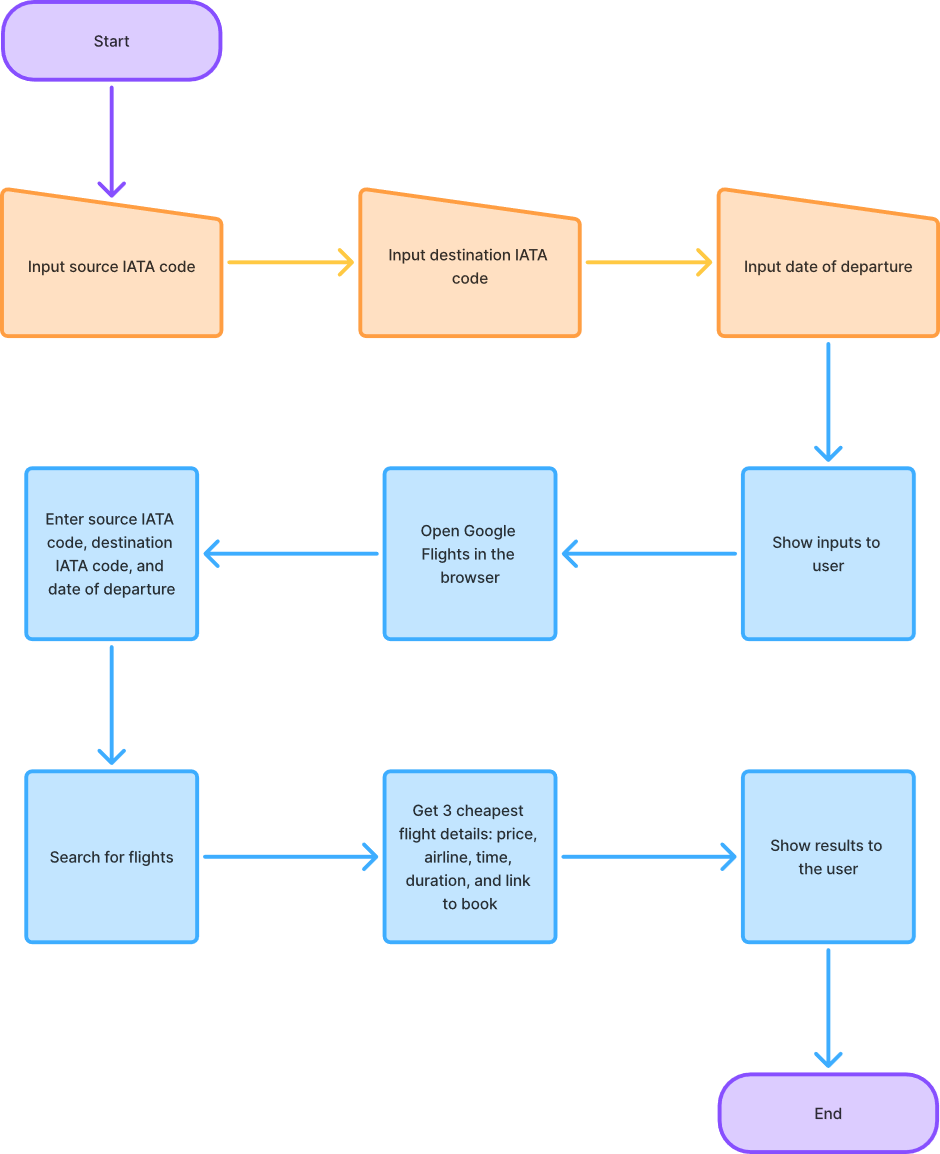
**4. PROJECT IMPLEMENTATION**

**4.1 Module Overview**

1. **Launch & Navigate**: Opens Chrome with constructed URL.
2. **Scrape Stage**: Loops over flight‑card elements, stores raw strings.
3. **Parse & Clean**: Regex to isolate ₹ value; split airline, times, duration.
4. **Rank & Select**: Sort list ascending; select first three.
5. **Output**: Display a message box.

**4.2 Tools & Technologies**

* Automation Anywhere A360
* Google Chrome on Windows 11
* Regex
* XPath

**4.3 Algorithm Details**

 **Input Stage** – Receive source airport, destination airport, and travel date from the user.

 **Navigation Stage** – Direct the bot‑controlled browser to the Google Flights results page for those inputs.

 **Collection Stage** – Gather all flight entries displayed on the page.

 **Processing Stage** – Extract fare, airline, times, and duration from each entry and sort the list by lowest price.

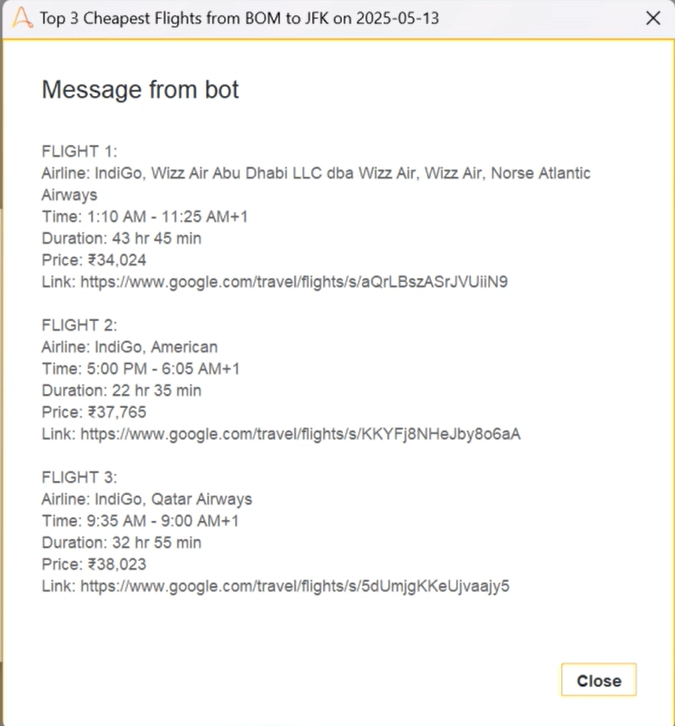
 **Selection Stage** – Keep only the three cheapest flights.

 **Output Stage** – Present the selected flight details to the user and end the session.

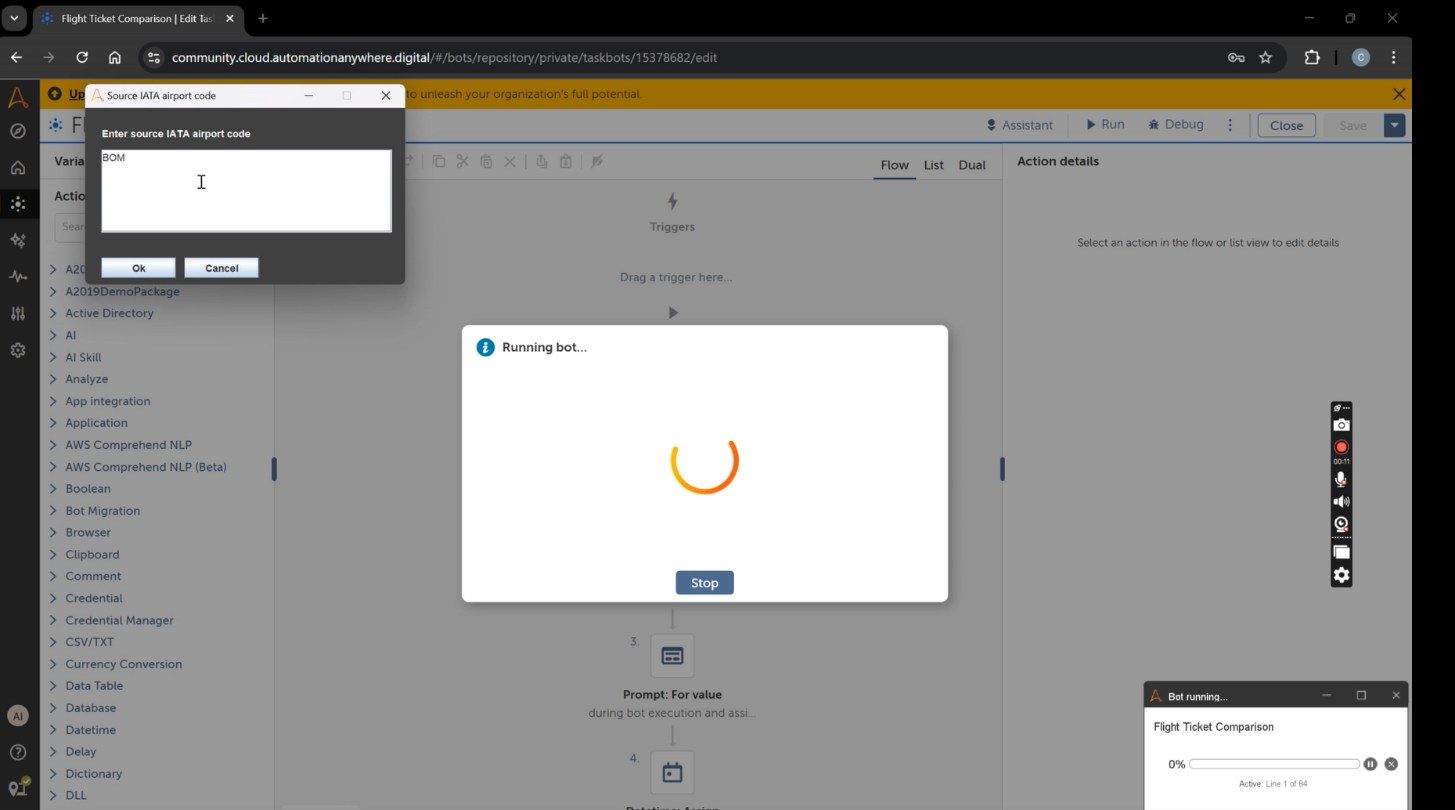
**5. RESULTS**

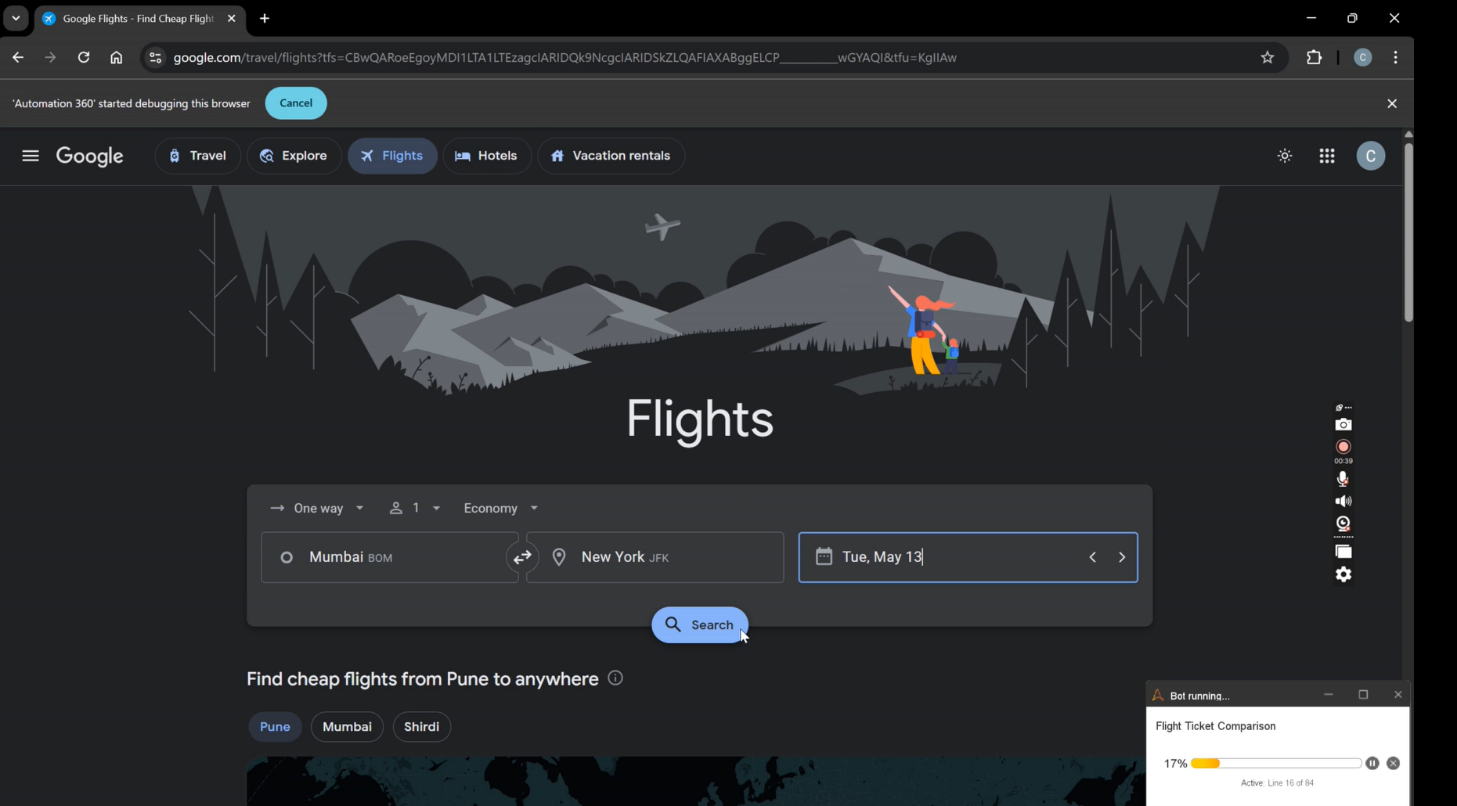
**5.1 Outcomes**

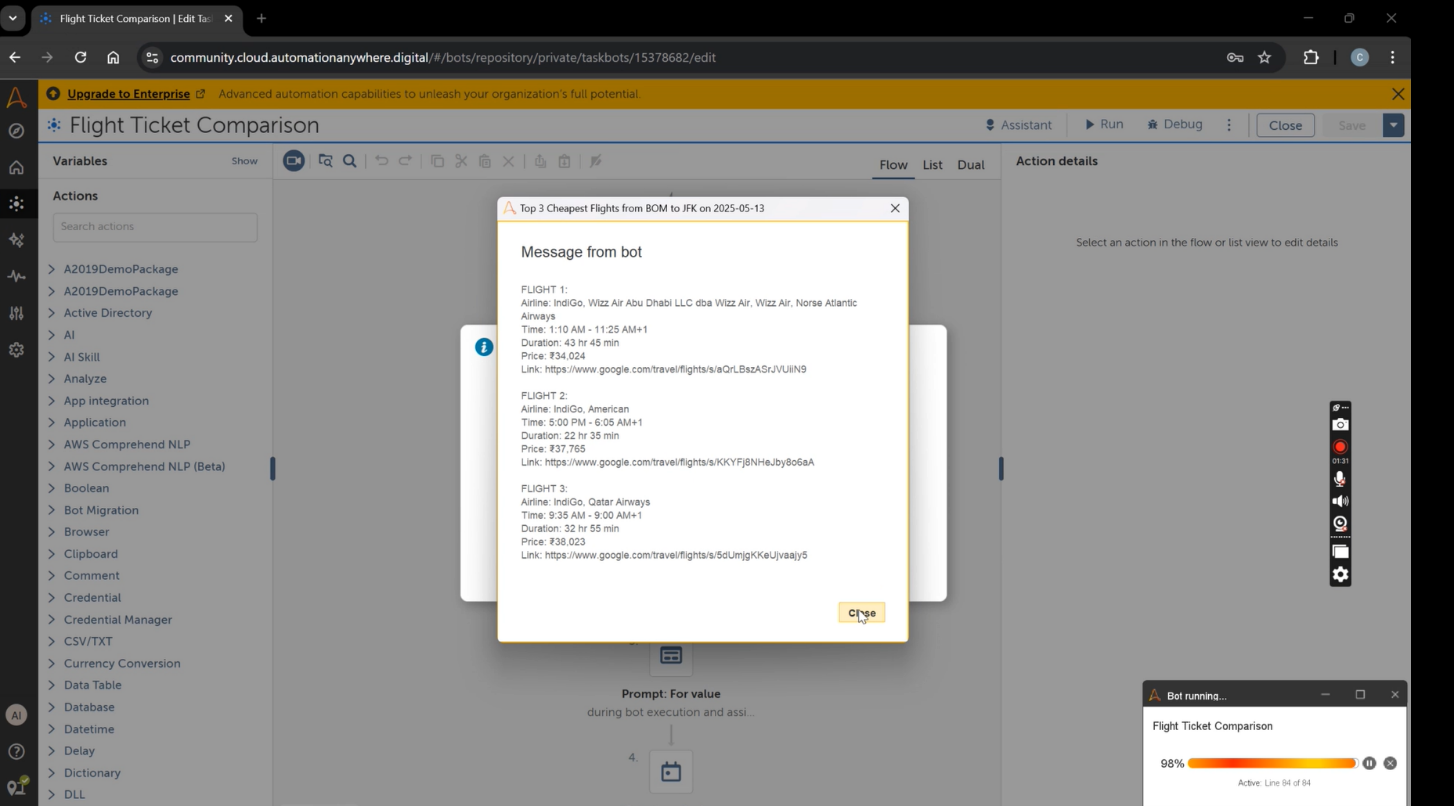
Bot returns message box:



**5.2 Screenshots**



****

****

**5.3 Metrics**

Execution time: 79 seconds

**6. CONCLUSIONS & FUTURE WORK**

**6.1 Conclusions**

The RPA bot successfully automates fare comparison, providing instant cheapest‑flight insights without API costs or manual effort.

**6.2 Future Work**

* Add return/multi‑city support.
* Build price‑drop alerts via email.

**6.3 Applications**

* Student travel desks
* Corporate travel cost‑cutting
* Personal trip planning

**REFERENCES**

1. Skyscanner API Docs, 2023.
2. Cleartrip Public API Reference, 2022.