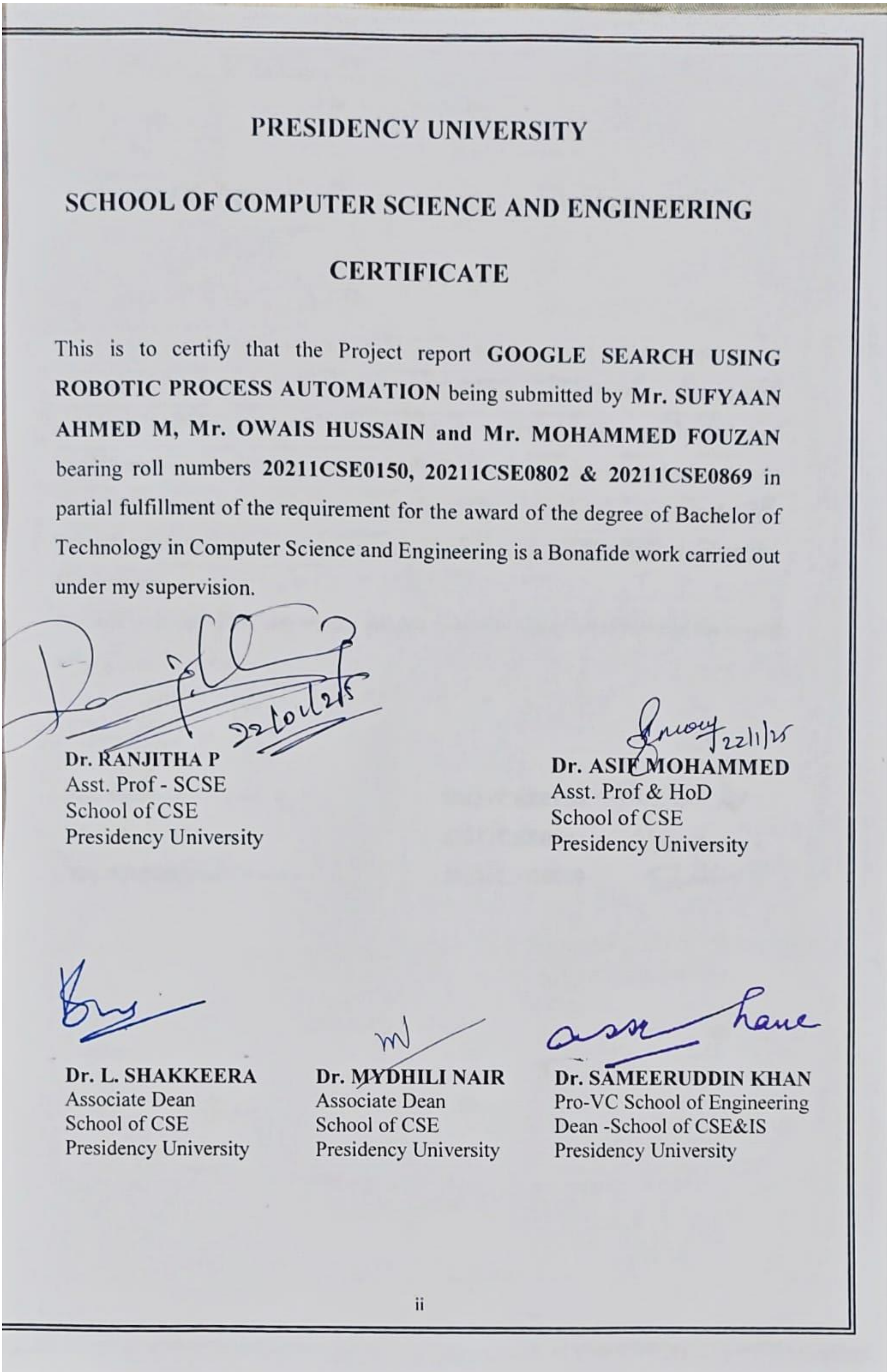


PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report **GOOGLE SEARCH USING ROBOTIC PROCESS AUTOMATION** being submitted by **Mr. SUFYAAN AHMED M, Mr. OWAIS HUSSAIN and Mr. MOHAMMED FOUZAN** bearing roll numbers **20211CSE0150, 20211CSE0802 & 20211CSE0869** in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a Bonafide work carried out under my supervision.

22/01/25

Dr. RANJITHA P
Asst. Prof - SCSE
School of CSE
Presidency University

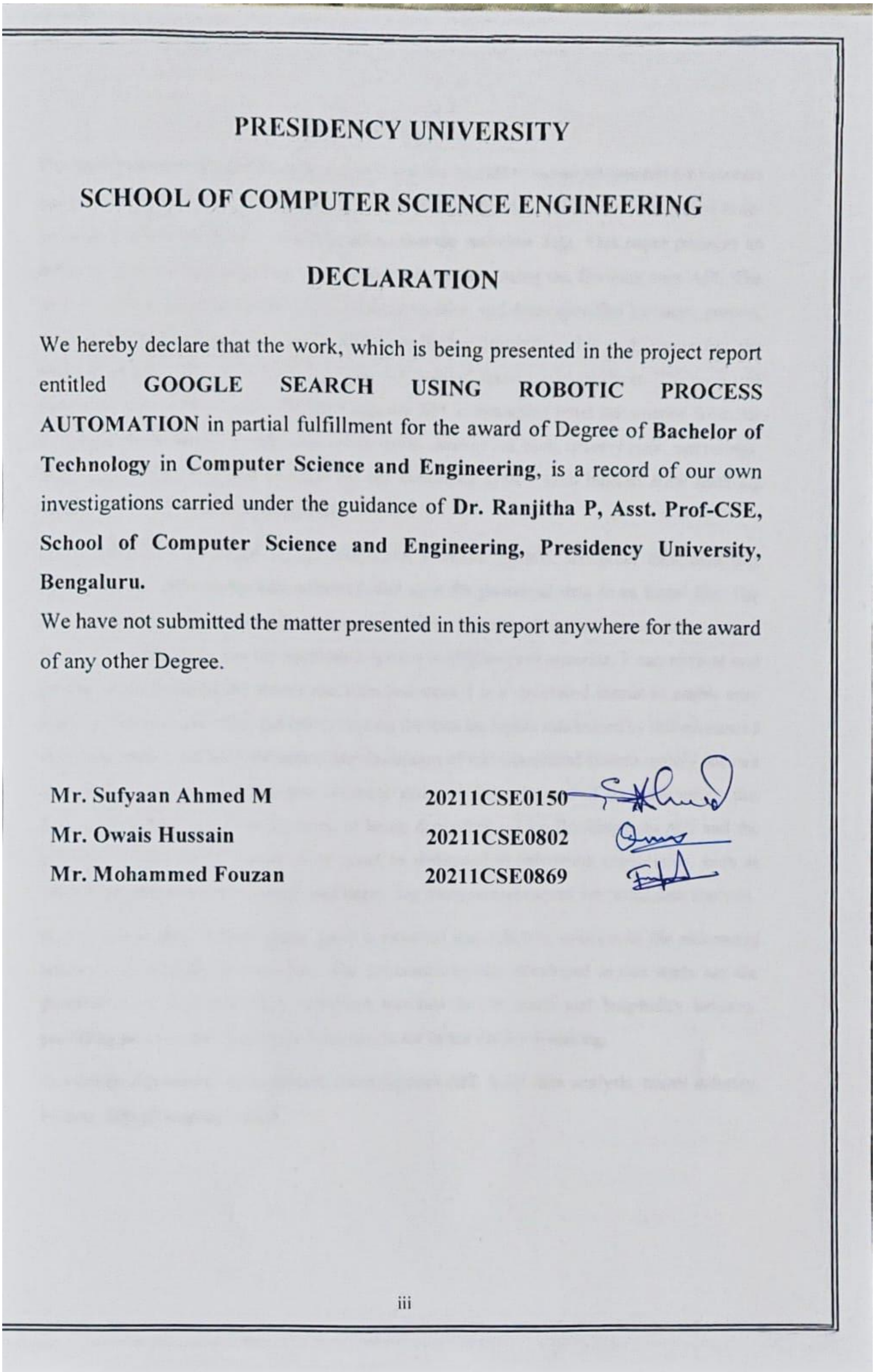
22/1/25

Dr. ASIF MOHAMMED
Asst. Prof & HoD
School of CSE
Presidency University

Dr. L. SHAKKEERA
Associate Dean
School of CSE
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-VC School of Engineering
Dean -School of CSE&IS
Presidency University



PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **GOOGLE SEARCH USING ROBOTIC PROCESS AUTOMATION** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. Ranjitha P, Asst. Prof-CSE, School of Computer Science and Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Mr. Sufyaan Ahmed M	**20211CSE0150**
Mr. Owais Hussain	**20211CSE0802**
Mr. Mohammed Fouzan	**20211CSE0869**

GOOGLE SEARCH USING ROBOTIC PROCESS AUTOMATION

A PROJECT REPORT

Submitted by,

Owais Hussain	**20211CSE0802**
Mohammed Fouzan	**20211CSE0869**
Sufyaan Ahmed. M	**20211CSE0150**

Under the guidance of,

Dr. Ranjitha P

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2025

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report **GOOGLE SEARCH USING ROBOTIC PROCESS AUTOMATION** being submitted by **Mr. SUFYAAN AHMED M, Mr. OWAIS HUSSAIN and Mr. MOHAMMED FOUZAN** bearing roll numbers **20211CSE0150, 20211CSE0802 & 20211CSE0869** in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a Bonafide work carried out under my supervision.

Dr. RANJITHA P
Asst. Prof - SCSE
School of CSE
Presidency University

Dr. ASIF MOHAMMED
Asst. Prof & HoD
School of CSE
Presidency University

Dr. L. SHAKKEERA
Associate Dean
School of CSE
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-VC School of Engineering
Dean -School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **GOOGLE SEARCH USING ROBOTIC PROCESS AUTOMATION** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. Ranjitha P, Asst. Prof-CSE, School of Computer Science and Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Mr. Sufyaan Ahmed M	**20211CSE0150**
Mr. Owais Hussain	**20211CSE0802**
Mr. Mohammed Fouzan	**20211CSE0869**

ABSTRACT

The rapid growth of the travel and hospitality industry has led to increased demand for accurate and up-to-date information about hotels. Traditional manual data collection methods are time-consuming, prone to errors, and often fail to provide real-time data. This paper presents an automated system for retrieving and analyzing hotel data using the Booking.com API. The system is set up to extract hotel information by location and dates specified by users, process this information, and save it to an Excel file for further processing, this study researches the issue of getting access to fast and accurate information about travel products. The proposed automatic system will employ the Booking.com API in extracting hotel information from the web page, which would include a hotel's ID, name, description, rank, country code, and ratings. The Python implementation is made for the developed system with built-in error handling capabilities and strong data processing.

Fetch destination ID based on the user-input location, retrieve the hotel data with that destination ID, process the data retrieved, and save the processed data in an Excel file. The system takes check-in and checkout dates input from the user, allowing them to choose travel dates, the results show that the automated system is efficient and accurate. It can retrieve and process hotel data with the utmost precision and store it in a structured format to enable easy analysis. The time and effort put into collecting the data are highly minimized by this automated approach, unlike manual approaches, the discussion of this automated system mainly focuses on benefits such as enhanced data accuracy and real-time retrieval of data. However, this system has a few limitations in terms of being dependent on the Booking.com API and the existence of rate limits. Future work could be dedicated to enhancing capabilities, such as integrating additional data sources and improving analysis techniques for better data analysis.

In conclusion, this research paper gives a practical and efficient solution to the automated retrieval and analysis of hotel data. The automated system developed in this study has the potential to revolutionize data collection methods in the travel and hospitality industry, providing accurate and up-to-date information for better decision-making.

Keywords: Automated data retrieval, Booking.com API, hotel data analysis, travel industry, Python, data processing, Excel

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameer Uddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair,** School of Computer Science Engineering & Information Science, Presidency University, and **Dr. SAMEERUDDIN KHAN** Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide and Reviewer **Dr. Ranjitha P, Asst ProfCSE**, School of Computer Science Engineering & Information Science, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar and Mr. Md Zia Ur Rahman** and Git hub coordinator Mr. **Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Sufyaan Ahmed M
Owais Hussain
Mohammed Fouzan

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

This project, "Google Search Using RPA," explains the inefficiencies and challenges involved in manually extracting structured data from Google searches. Traditional search processes require a lot of manual effort, which is prone to errors and time consumption, especially in large-scale data. RPA, combined with APIs, automates the workflow, ensuring accurate and efficient data collection. The system focuses on retrieving hotel details, such as names, ratings, and descriptions, using Booking.com's API. The project integrates RPA to handle user queries dynamically and output data into a structured Excel format. It begins with an introduction that discusses the relevance of automation, how APIs improve reliability, and the time-saving nature of the project while ensuring the accuracy of the data. In addition, it defines the scope of the project, emphasizing its applications in hospitality, business intelligence, and more.

1.1 Overview of Automating Google Search for Data

The project aims to automate the process of extracting valuable data from Google search results, allowing for faster, more efficient, and scalable data collection. This automation is useful for various applications, including market research, competitive analysis, trend analysis, and gathering customer insights. By automating the search and data extraction process, businesses, researchers, and analysts can access large volumes of structured data with minimal effort, avoiding the time-consuming manual searching and data organization.

- **Objective and Key Goals**

  The objective of this project is to enable the automatic retrieval of relevant information from Google search results for any query. The user may query for information, such as "best hotels in Paris" or "latest trends in machine learning." The automation process will thus increase the access of real-time and accurate structured information from web-based sources to the analysis process in order to draw conclusions.

- **Automation Process**

  - ➢ User Input: This would be the first input as a search query from the user. It can be the location, an industry trend, or product search.
  - ➢ Google Search Execution: The system executes the Google search automatically through web scraping tool or API. This search is done by the

system through the help of web scraping tools such as Selenium or Beautiful Soup or through the official Google API, Custom Search API.

- ➢ Data Extraction: The system will extract all relevant information from the search results, including:

- **The titles and URLs of the search results.**
  - ➢ Meta descriptions or snippets for each of the search results.
  - ➢ Structured data like product prices, hotel availability, or reviews on the linked pages.
  - ➢ Data Processing: Once extracted, the data is cleaned and processed to eliminate unnecessary information, filter out duplicates, and format it in a useful manner.
  - ➢ Output: The collected data is stored in structured forms like Excel files, CSV files, or databases for analysis and reporting.
- **Problems**
  - ➢ Despite the numerous benefits of scraping Google search, the project will still face some challenges:
  - ➢ Google Restrictions: Google's terms of service do not allow scraping its search results. The project needs to abide by these terms of service, and it may need to use the official APIs, for example, Google Custom Search API.
  - ➢ Captcha and Rate Limiting: Google will block requests from automated systems if they find that the activity is unusual or repetitive. For this reason, developers use the technique of rotating IPs or use proxies to circumvent it.
  - ➢ Data Consistency: It can be a daunting task to clean and extract data from search results, especially on dynamic or JavaScript-heavy web pages. Error handling and data validation become necessary for the accuracy.
  - ➢ Applications and Use Cases
  - ➢ Automating Google search data extraction has a lot of applications in:
  - ➢ Market Research: To gather data on competitors, price lists of products, and promotion strategies to conduct competitive analysis.
  - ➢ SEO Analysis: Scrutinizing search rankings, snippets, and content that come out through the search to fine-tune SEO strategies.
  - ➢ Customer Reviews: Aggregating reviews from multiple websites to conduct sentiment analysis and aggregation of feedback.

- Trend Analysis: Automating the collection of industry trend insights by scraping news articles, blogs.
- It helps streamline the collection of large quantities of information through automated Google search. One can collect structured data from search results by using APIs and web scraping tools within less time
- compared to humans doing the task. This is very helpful for industries requiring a massive dataset, and on-time access might help them get an upper hand over those competitors in such sectors as market research, SEO, and trend analysis. By overcoming the CAPTCHA and rate limiting challenges, and by ensuring data accuracy, businesses and individuals can unlock valuable insights more efficiently and effectively. The information is rendered and fetched using a headless brow ser like Selenium.

1.2 Requirement to Automate Hotel Data Extraction

The travel and hospitality industry is increasingly data-driven. More information about where to go, when to go, and what to do there should be at travelers' fingertips for the perfect trip planning. Extracting hotel data manually—names of hotels, prices, ratings, amenities, and reviews—is time-consuming, error-prone, and not efficient. Therefore, the need to automate the process of extracting hotel data to improve the process: faster, accurate, and scalable solutions to businesses, acquiring information about a hotel manually involves going through Internet sites, summarizing the information, and making it available. Sifting through vast volumes of data from different locations is very time-consuming and may contain human copying or recording errors. The information also needs to be updated frequently due to the dynamic nature of the hotel prices, their availability, and reviews. Besides, businesses or researchers processing data for various regions are burdened by the amount of information they have to process and standardize. The above challenges thus justify why manual methods are not feasible in today's fast-paced digital environment.

Automation is a valuable resource for travel agencies, price comparison sites, hotels, market researchers, and even in academic research. Travel agencies and price comparison sites can make use of automation to give clients up-to-date options for hotels and prices.

Hotels can follow competitors' rates to use in dynamic pricing, while researchers can analyze tourism and hospitality trends and patterns.

1.3 Role of API in Improving the Data Retrieval

Application Programming Interfacing (APIs) play a transformative part in improving the way information is collected, prepared, and utilized over different businesses. Within the travel and neighborliness segment, APIs have ended up vital for computerizing information recovery forms, giving organized, real-time get to to important data such as inn points of interest, costs, accessibility, and audits. By bridging frameworks and stages, APIs empower businesses, designers, and analysts to streamline their operations and center on conveying esteem, APIs serve as middle people that permit computer program applications to associated with databases or administrations. Not at all like manual strategies of information collection, which include exploring websites and extricating data through web scratching, APIs give a coordinate and solid channel to get to information from stages such as Booking.com, Expedia, and TripAdvisor. These APIs return information in organized designs like JSON or XML, which makes handling and integration direct.

APIs give momentary get to to energetic information. For occurrence, lodging APIs can return overhauled room accessibility, costs, and client reviews in real time. This can be basic in businesses where data changes regularly, such as travel, where accessibility and estimating change, APIs guarantee that the recovered information is clean, organized, and reliable, disposing of the blunders and inconsistencies frequently experienced with manual information scratching. They permit designers to center on building applications without stressing around information quality, Most APIs bolster customized parameters, such as sifting information by area, budget, check-in/check-out dates, or particular civilities. This guarantees that clients get as it were important information custom fitted to their needs.

APIs coordinated easily with other apparatuses, making them appropriate for end-to-end forms like information examination and visualization. For occasion, information recovered through APIs can be specifically prepared utilizing Python libraries or visualized in instruments like Scene, making a difference lodgings screen competitor estimating to alter their rates powerfully, Conveying personalized inn suggestions based on client inclinations. APIs have revolutionized the information collection handle, giving businesses with productive, versatile, and dependable arrangements. In businesses like travel and neighborliness, where information changes quickly, APIs guarantee that businesses stay competitive by empowering real-time decision-making and consistent robotization. By leveraging APIs, companies can improve efficiency, provide superior client encounters, and open unused openings for advancement.

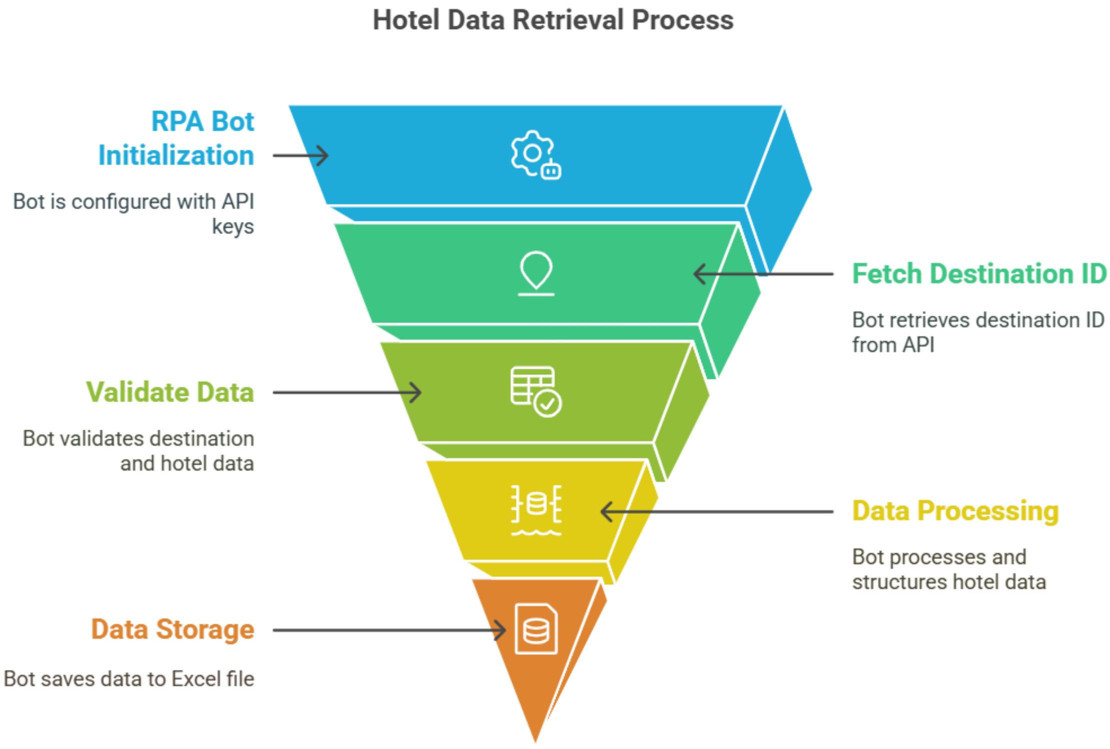

Hotel Data Retrieval Process
RPA Bot Initialization
Bot is configured with API keys
Fetch Destination ID
Bot retrieves destination ID from API
Validate Data
Bot validates destination and hotel data
Data Processing
Bot processes and structures hotel data
Data Storage
Bot saves data to Excel file

Figure1.3: Data Retrieval

1.4 RPA for Structured Data Extraction

Mechanical Prepare Mechanization (RPA) has ended up progressively significant within the era and administration of organized information, advertising a transformative approach for organizations looking to computerize monotonous assignments, improve precision, and move forward generally operational proficiency. Organized information alludes to data organized in a characterized organize, such as tables, spreadsheets, or databases, making it less demanding to look, analyze, and prepare. RPA can streamline different forms including organized information, and its applications are tremendous and impactful, one of the essential benefits of RPA in organized information era is information extraction. Numerous businesses work with tremendous sums of information put away over numerous frameworks or stores. RPA instruments can computerize the extraction of organized information from these sources without the required for manual mediation. For occurrence, information can be pulled from frameworks like Client Relationship Administration (CRM) program, venture asset arranging (ERP) frameworks, or spreadsheets. Robotizing this handle guarantees that information is collected effectively, precisely, and reliably, which is particularly basic in large-scale information operations. Another vital angle of RPA is data entry, a assignment that's as often as possible time-consuming and inclined to human blunder when done physically. RPA robots can input organized information into shapes, databases, or reports consequently. This diminishes the probability of blunders, guarantees reliable organizing, and speeds up information handling. Whether it's entering client subtle elements into a CRM or overhauling stock information into a database, RPA kills manual blunders and minimizes the time required for these dreary errands.

Moreover, RPA can help in information change, which is the method of changing over information from one organize to another, such as from a CSV record to an Exceed expectations spreadsheet or between diverse database frameworks. This change regularly requires complex rationale and monotonous steps, which can be effortlessly mechanized with RPA apparatuses. By mechanizing information change, RPA guarantees that information remains intact and precise, which it is prepared productively over systems.RPA is additionally profitable in data integration. Numerous organizations work with different frameworks that store information completely different designs or stages. RPA can bridge the gap between these frameworks, exchanging organized information consistently over stages. This integration guarantees that all frameworks are synchronized which information remains exact, up-to-date, and effectively open for commerce forms, in expansion, RPA empowers mechanized detailing. RPA instruments can drag organized information from different sources, organize it, and produce reports, whether for inside partners or outside clients.

1.5 Scope and Applications of the System

Mechanical Prepare Robotization (RPA) plays a urgent part in overseeing and producing organized information over different businesses. Its scope amplifies to mechanizing assignments that include information extraction, change, passage, approval, and detailing, advertising critical benefits such as expanded productivity, exactness, and adaptability. RPA's capacity to handle monotonous assignments decreases human blunders and quickens trade operations, making it crucial device for organizations managing with huge volumes of organized data. One of the essential applications of RPA is in information extraction, where it mechanizes the recovery of organized information from different sources like databases, spreadsheets, and outside frameworks. This kills the require for manual passage, upgrading precision and speed. For case, RPA can consequently extricate budgetary information from bank articulations or client points of interest from CRM frameworks. This mechanized information extraction guarantees businesses spare time and decrease the chance of human errors. Data passage could be a common errand in most organizations, frequently including dreary and time-consuming exercises. RPA computerizes this handle, guaranteeing that organized information is precisely entered into frameworks, shapes, or databases. It can moreover perform information overhauls in real-time, guaranteeing that records are synchronized over frameworks. In human assets, for occasion, RPA can overhaul worker records consequently, guaranteeing information is continuously current without manual mediation.

RPA exceeds expectations at information change, changing over information from one arrange to another, such as from CSV records to Exceed expectations or between distinctive database frameworks. This can be particularly valuable amid framework movements or when uniting information from different sources. By computerizing this prepare, RPA guarantees information consistency and decreases the time went through on manual information taking care of, making it simpler to oversee expansive datasets. Ensuring information judgment is pivotal, and RPA can be modified to naturally approve organized information. It can check for blunders, lost areas, or irregularities in data before it is entered into frameworks. Typically, particularly imperative in businesses like back, where precise information is fundamental for compliance. For illustration, RPA can approve exchange information to guarantee it meets bookkeeping standards. RPA enhances the administration of organized information by robotizing assignments such as extraction, passage, change, approval, and announcing. This leads to critical advancements in operational productivity, information exactness, and adaptability, profiting businesses like fund, healthcare, retail, and more.

1.6 Issues with Manual Data Extraction

Manual information extraction, whereas regularly a vital errand in numerous organizations, comes with a assortment of challenges that can essentially ruin productivity, precision, and in general efficiency. These challenges can result in expanded operational costs, delays in decision-making, and mistakes that influence the quality of the information being utilized for trade operations. One of the foremost critical challenges of manual information extraction is the time it expends. Extricating information from different sources such as records, spreadsheets, or databases requires a impressive sum of time, especially when managing with expansive volumes of data. This manual handle can delay basic trade operations, anticipating organizations from reacting rapidly to changes or making convenient choices.

Human mistake could be a common issue in manual information extraction, and indeed the foremost experienced representatives can make botches. Mistakes such as information passage botches, lost information, or off base translation of data are visit when assignments are done physically. These botches can compromise the quality of the information, driving to inaccurate announcing or decision-making, particularly in businesses that depend on tall information precision, such as back, healthcare, and legitimate divisions.

Manual information extraction regularly leads to information irregularities, particularly when it includes numerous individuals dealing with the same information. Diverse groups or traditions may be utilized by distinctive group individuals, coming about in errors when the information is put away or utilized afterward. This need for consistency complicates information administration, requiring extra exertion to accommodate and standardize the data sometime recently it can be utilized effectively. Manual information extraction regularly leads to issues with information quality. Since the method is subjective, it is inclined to irregularities in how information is translated, entered, or approved. These irregularities can influence information unwavering quality, driving to destitute decision-making. Wrong information can have noteworthy results, especially in compliance-heavy businesses like fund or healthcare, where keeping up precise records is basic.

In businesses with strict administrative requirements, manual data extraction can complicate endeavors to preserve compliance. Guaranteeing that extricated information meets legitimate and industry measures can be troublesome when forms are manual.

CHAPTER-2

LITERATURE SURVEY

Automation has now become a tool for handling big data and huge data collection with extensive organization in the modern age. The older methods of extracting data, including manual searches and basic web scraping, are greatly limited and inefficient in today's fast-paced world. This gives rise to a need for sophisticated and automated systems.

2.1 Existing Solutions

- Manual Search:
  Manual data extraction uses human effort to browse through the search results to identify information of relevance and copy it into usable format. This approach has the advantage of accuracy at specific points; however, it is profoundly time-consuming and inefficient especially when the data is considerable in volume. Furthermore, humans are wired to make errors and also replicate information differently, which will impact the validity of the collected data. For example, gathering contact numbers or business listings from several website pages manually creates human fatigue in addition to many transcription errors.

- Basic Web Scraping:
  Web scraping can be done automatically with the aid of Beautiful Soup and Selenium. There are many benefits in using the two tools that have been implemented for years in web scraping tasks because they have simplicity and versatility. However, there are difficulties that come along with modern or dynamic websites. Most web pages employ JavaScript for dynamic loading of content, which makes it hard to access the desired information through basic web scraping tools. Additionally, websites use captchas and apply rate limits to identify and cancel scraping, adding complexity in terms of IP rotation or even using headless browsers.

2.2 Problems Facing Current Approaches

In traditional approaches, the main problem is that data retrieved is in unstructured form. Search results are not in a format that is easily usable, and therefore often need further cleaning and processing to be useful for analysis or application. Another challenge is dynamic content, where the reliance on JavaScript to render data makes scraping more difficult. Tools need to be able to execute JavaScript or navigate DOM structures dynamically, which increases computational overhead and development time. Last but not least, the rate limits and captchas put in place by websites protect against abuse but present major barriers to automation tools.

2.3 Improvements in RPA and APIs

RPA is a strong alternative to normal scraping methods because it imitates human activities. In contrast with most scraping tools, with RPA, solutions interact with web pages in a manner similar to that of humans, clicking buttons, filling out forms, and downloading visible data. This method allows for higher precision and also cuts down on the chance of getting blocked on websites, another reliable source of structured access to data is through APIs such as Rapid API. APIs give developers the capability to fetch particular information directly from the service provider, eliminating the need to parse HTML or handle complex website structures. This would eliminate issues such as dynamic content and captchas. The use of RPA and APIs will present a scalable and efficient method of data extraction, providing structured outputs and lowering development and maintenance efforts, by leveraging these advancements, the project overcomes the limitations of traditional methods, ensuring efficient and reliable data extraction in a structured format, tailored to specific needs.

CHAPTER-3
RESEARCH GAPS OF EXISTING METHODS

Although automation tools and techniques have made tremendous advancements in data extraction and organization, the following research gaps are still pending. These research gaps indicate areas where current methods are lacking, and further innovation is needed to meet the needs of real-world applications. Here are the key research gaps and their implications for the field of automated data handling.

3.1. Lack of Structured Output

- **Problem**
  Traditional data extraction through manual search or simple web scraping often ends up in providing unstructured data. For example, Beautiful Soup for web scraping stores the raw data from HTML output as it brings all the redundant tags, scripts, and unwanted data. Subsequently, after extraction, most of the outputs end up getting extensively cleaned to become usable for use. For example, cleaning will include removing duplications, addressing missing values, and formatting of data into the desired structured file type such as CSV or Excel.
- **Gap**
  While there are many tools to extract data from websites, not many of them offer end-to-end solutions which provide directly usable, structured output specific to particular use cases. For example, contact details or addresses extracted from search results typically require some amount of additional programming effort to parse and organize. An effective system which automatically does both extraction and organization has been rarely explored.

3.2. Problems with Dynamic and JavaScript-Driven Content

- **Problem**
  The increasing use of JavaScript to dynamically render web content poses significant challenges for data extraction. Most modern websites load data only when users interact

with specific elements (e.g., scrolling or clicking), making it inaccessible to traditional scraping tools. Scrapers often struggle to interact with or extract data from these dynamically generated elements, resulting in incomplete or inaccurate datasets.

- **Gap**

  The currently prevalent tools do not have an efficient way of dealing with JavaScript-driven content without resorting to a computationally expensive solution like a headless browser. While these browsers can perform JavaScript, they are too resource-intensive to be used for any considerable or real-time applications. What is required is a lightweight effective solution that can smoothly handle dynamic web content.

3.3. Rate Limits and Anti-Bot Mechanisms

- **Problem**

  Webs use a number of anti-bot mechanisms in the form of captchas, rate limits, and IP blocking to sniff out and limit activities of automated scrapes. Scraper applications have tried various bypass techniques in the form of IP rotation, proxy servers, or captcha-solving services, which usually are unreliable and fail most of the time. Repeated offenses will attract permanent bans or lawsuits.

- **Gap**

  There is limited research on developing ethical and scalable solutions that can bypass these restrictions while being in line with terms of service and maintaining data accuracy. There is a great need for methods that respect website policies and comply with legal requirements while ensuring seamless data extraction.

3.4. Data Relevance and Filtering

- **Problem**

  Basic web scraping tools tend to extract all visible data on a webpage without making any distinction between relevant and irrelevant information. This indiscriminate approach results in datasets that contain noise, such as duplicate or extraneous data, which increases the overhead during the cleaning and processing phases. For instance, extracting information about hotels may include irrelevant content like advertisements or unrelated links.

- **Gap**

  Current methods lack intelligent filtering mechanisms that allow users to specify parameters or criteria for relevance during the extraction process. Developing systems with built-in filtering capabilities could significantly reduce post-extraction processing time and enhance the usability of the data.

3.5. Limited Integration with APIs

- **Problem**

  APIs like Rapid API provide structured and reliable data access, but the integration process is usually complex. Many APIs also have restrictions such as rate limits, limited free access, or subscription fees, which can limit their adoption in small-scale projects or by non-technical users.

- **Gap**

  Tools exist that rarely leverage the strengths of RPA in combination with APIs. This usually leads to lost opportunities for building scalable systems which extract data in an efficient, reliable manner, with minimal effort.

3.6. Scalability and Domain-Specific Applications

- **Problem**

  Many data extraction tools are not scalable for handling large datasets or adaptable to specific domains. For instance, a scraper that is designed for extracting hotel information would need to be significantly customized to work for educational institutions or retail businesses. This lack of versatility limits the application of existing solutions in broader contexts.

- **Gap**

  There is a need for flexible systems that can scale across domains while being accurate and reliable. Research should be focused on creating modular and adaptable solutions that can cater to diverse industries without requiring extensive reconfiguration.

3.7. Ethical and Legal Concerns

- **Problem**

  Web scraping often raises ethical and legal concerns, particularly when accessing restricted or private data. Many tools operate in a gray area, where their actions may violate website terms of service or data protection regulations like GDPR. This creates risks for developers and users, including potential legal consequences.

- **Gap**

  There is scarce research on creating compliant and ethical data extraction techniques that adhere to website policies and respect the privacy of users. Exploring open and legally defensible approaches might encourage broader use of automation tools.

CHAPTER-4
PROPOSED MOTHODOLOGY

4.1 Requirement Analysis

The first phase of the proposed methodology includes carrying out an analysis that helps determine what is exactly required from the system. In other words, it will help to figure out which data points need to be extracted as a result of the user's query. For example, in the case of a query such as "hotels in Goa," the data points might be names of hotels, contact numbers, email addresses, physical addresses, and links to their websites. Each search query will also require specifying domain-specific parameters pertaining to hotels, schools, or restaurants, for example. The system shall be designed with a modular architecture, which will enable its adaptability for any number of domains. Being modular will, therefore, ensure that the solution can be scaled and adapted with minimal overhauls according to the requirements in various use cases. The output should be formatted to enable analysis, and it is most commonly in the format of an Excel spreadsheet or a CSV file.

4.2 Data Collection

This is the step where data is collected using RPA, API integration, and web scraping for the needed information.

- Robotic Process Automation (RPA):
  RPA tools, including UiPath, Automation Anywhere, or Python-based libraries like Selenium, execute the automation process of web browsing. These tools make human-like interactions, like the inputting of keywords in Google search queries, navigation to the results, and clicking of links based on relevance. RPA makes sure the entire process, from search to data collection, is performed; it has less human intervention and can extract data from multiple pages within the least effort.

- Integration with APIs:
  Structured and reliable data can be achieved using APIs such as Rapid API or Google Places API. Direct access to structured data like ratings, contact information, and business descriptions can be done through APIs instead of scraping web pages. Data becomes more consistent because it avoids the problems that occur with dynamic

content or anti-scraping mechanisms. The use of APIs ensures that the data collection process is more efficient and accurate, especially when scraping web content is difficult or legally restricted.

Endpoint	Purpose	Method
/api/v1/hotels/searchDestination	Fetch destination ID	GET
api/v1/hotels/searchHotels	Fetch Hotel data	GET

Table 4.2: API Endpoint

- Web Scraping:

  In the case where APIs are not available or provide inadequate information, web scraping is often taken into consideration. Using the Python library Beautiful Soup and Scrapy, it will be applicable to simply use the data obtained in the HTML content of the results pages of the search. It is possible to extract only the important details from the use of HTML as its parsing of structural code.

Attribute Name	Description	Data Type
Hotel ID	Unique identifier for the hotel	String
Name	Name of the hotel	String
Description	Brief description of the hotel	String
Ranking Position	Ranking position of the hotel	Integer
Rating	Overall rating of the hotel	Float

Table 4.2: Data Attributes

4.3 Data Cleaning and Filtering

After gathering the data, cleaning and filtering the extracted information is the next step. This is a crucial step to ensure that the quality and usefulness of the data are preserved.

- Duplicate Elimination: The system will eliminate duplicate entries by identifying them. For instance, if there are several listings of the same hotel retrieved, then only one entry will be left.

- Validation: The system will validate the extracted data by authenticating in terms of proper formats. That is, an email should have the right structure, and a phone number should fit into the regional pattern. The overall errors result in the validation of this dataset.
- Relevance Filtering. Data quality can be improved further through relevance filtering wherein only information meeting certain user defined criteria are retained in the database. In example, this involves elimination of noise factors like irrelevant adverts and extraneous links ensuring only pertinent information remains in data points.

4.4 Data Organization

The cleaned data would then be compiled in a structured form, such as an Excel file or CSV. This way, when extracting data such as hotel names, contact numbers, and addresses, it is easily interpreted for further analysis as it will go into its columns. Libraries, such as OpenPyXL or XlsxWriter in Python, would be applied to create the Excel file as well as ensure the output format is user-friendly and visually aesthetic.

4.5 Ethical and Legal Compliance

The proposed system is designed to emphasize ethical considerations and legal compliance, such as GDPR. The system will mainly depend on APIs that provide authorized and structured access to company data. This minimizes the risks of web scraping, especially on sensitive or restricted data. The RPA workflows will also be responsible for honoring websites' terms of service in such a way that no scraping activities violate any legal or ethical limits.

4.6 Testing and Validation

This system will undergo rigorous testing to ensure it becomes reliable and effective. Some of the crucial testing areas include:

- Accuracy: The extracted data has to be very close to what is contained in source websites.
- Scalability: The system will be tested to ensure large amounts of data are handled without performance issues.
- Usability: The Excel output will be tested for clarity and easy interpretation to meet

user requirements.

4.7 UI Development

An easily navigable interface will be built for operation of the system with users who have non-technical skills. Such an interface shall support entering queries on hotels in Goa or in similar phrases for that matter. A field of user preferences toward desired output would also be possible so that they would receive results as requested-let's say in an Excel sheet format-and it is guaranteed to come through, which reduces efforts of operation at hand for a user as much as possible.

4.8 Deployment

Finally, the system will be deployed as a standalone application or integrated into existing workflows. It will be designed to work across multiple operating systems, including Windows, macOS, and Linux. A help section will be included to guide users through the process of using the system and troubleshooting common issues, ensuring that the system is accessible and easy to use for all users.

CHAPTER-5
OBJECTIVES

The project objective is mainly the automation of extracting and organizing structured data from Google search results by using RPA and APIs. It aims at coming up with a system that can autonomously extract valuable information such as hotel names, contact numbers, email addresses, and physical addresses depending on user-defined queries, then store these in a structured, easy-to-use format.

5.1. Data Collection Automation

The first and most important goal is to automate the process of data collection. Traditionally, users manually search for information on search engines like Google and extract data, which can be time-consuming and inefficient. This project seeks to automate that entire process. By using RPA tools like Selenium and Automation Anywhere, the system will autonomously perform searches on Google based on user queries, such as "hotels in Goa" or "restaurants near me

5.2 Ensure Data Accuracy and Relevance

The system should ensure that only the most accurate and relevant data are captured in the right format. Its primary objective must be the effective mechanism for filtering and cleaning data to eliminate all irrelevant, duplicate, or wrong entries. This will be achieved in the system through:

- Remove duplicates: The system will remove multiple entries of the same business usually one for a hotel on different listings, therefore giving a clean dataset.
- Relevance: This system will eliminate advertisements, reviews, and other links that contain information irrelevant to the business searched.
- Data validation: For example, it will validate email addresses and make them in proper format or working. Similarly, phone numbers and URLs will be validated whether they are functional or not.

5.3 Export Data into a Structured Format

It would then ensure organizing the extracted data into structure that could easily be accessed for analysis purposes. One of the core objectives in this step would be ensuring the extracted data exported into one widely used format: an Excel sheet or a CSV file, The system will Organize the data into rows and columns, where each column will represent a specific data point, such as hotel name, contact number, email, address. Ensure that the structure is clear and user-friendly so that non-technical users can easily navigate the spreadsheet, Use Python libraries like OpenPyXL or XlsxWriter to create and save the data into Excel files, ensuring that the output is professional and formatted for usability. Users can easily analyze the data or use it for further applications such as marketing, research, or business development through exporting the data into a structured format.

5.4 Integrate APIs for Enhanced Functionality

Rapid API or Google Places API. APIs that are integrated, such as Rapid API and Google Places API, can supply reliable, structured data to be used along with web scraping that will further expand the details retrieved about a business. Integrating APIs, the system would: Get further information than available through a search on Google for example: rating, review, business hours, pricing, and much more. It offers better insight into the businesses these users are looking for, so the system is stronger and more useful to the users, It has reduced the necessity of depending on web scraping sometimes that is inconsistent and unmanageable due to dynamic content and rate limits.

5.5 Enhanced User Experience

The system needs to be user-friendly and easily understandable even by users with a minimal technical background. An important goal is to create a simple, easy-to-use interface that lets the user easily enter search queries and get their answers in a structured format. Allow users to enter queries such as "restaurants in New York" and select output format, say, Excel, CSV.Minimum number of steps needed to retrieve the data so the user can find the information required with minimal efforts. A wide range of users from small business to individual researcher will be able to access the system with a smooth user experience which increases the practicality and adoption.

5.6 Demonstrate Scalability

The system should be scalable. Scalability refers to the ability of a system to process large volumes of data and to support different types of queries in various domains. The system may be used to query hotels, schools, restaurants, or any other local business. The system must be able to adapt to various datasets. The system will ensure scalability through the following measures: The system should be able to process thousands of queries concurrently without losing its performance. Adapt to different domains with minimal modifications, so that users from different industries can use the tool for their specific needs. Handle queries from different geographical locations, providing localized results based on the user's query.

CHAPTER-6
SYSTEM DESIGN & IMPLEMENTATION

6.1 System Architecture

The system follows a modular architecture that includes the following modules:

- Input Module: Captures the user's queries, for instance, "hotels in Goa."Accepts specific parameters including output format like Excel or CSV, and domain preference, for instance, hotels, schools, or restaurants. There should be a friendly interface that can make it look simple and friendly to non-technical users.
- Data Extraction Module: Robotic Process Automation (RPA) is utilized to automate the Google search, and interacting with web pages. Integrate APIs such as Rapid API or Google Places API to fetch structured data where available. Use web scraping tools like Beautiful Soup and Selenium to fetch data from search results or specific websites.
- Data Cleaning and Filtering Module: Process raw data for removing duplicates and irrelevant entries. Validate data for accuracy and consistency, ensuring fields like email addresses and phone numbers meet proper formats.
- Data Storage and Organization Module: It organizes the cleaned data into rows and columns, where each column is assigned a specific data point, such as hotel name, phone number, or email.

  It saves the structured data into user-specified formats like Excel or CSV using libraries such as OpenPyXL or XlsxWriter.
- Output Module: The module will deliver the final data file to the user in the format selected. It ensures that the output is easily understandable, with clear headers and formatting for easier usability.

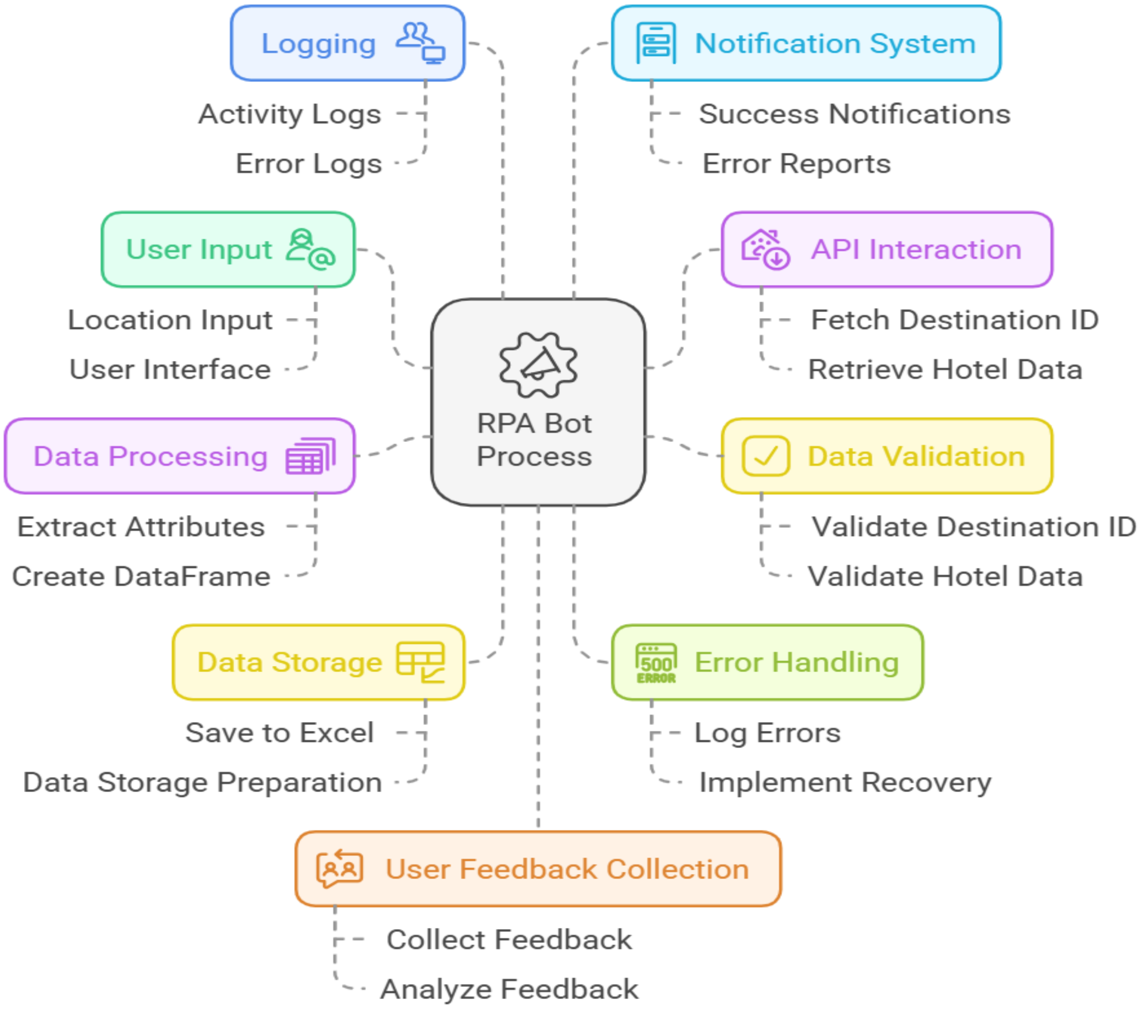


Figure 6.1: System Architecture

6.2 Implementation Steps

- Step 1: Requirement Gathering

  Gather the need of the users such as type of data, level of accuracy required, and output format, find suitable APIs and libraries to extract and organize data

- Step 2: Development of the Input Module

  Use frameworks such as Tkinter or PyQt to create a user interface for Python Provide input to search for a query and options to choose preference in extracting the data and desired output

- Step 3: Automation of Data Extraction
  - Implementation of RPA

Use Selenium or any other RPA tool to mimic typing of search keywords, clicking on the links, and moving through different web pages.

- API

  Access APIs, for example Rapid API or Google Places API that would retrieve the structured data on ratings, contact details, business descriptions, and many more.

- Web Scraping

  Use Beautiful Soup or Scrapy and scrape data straight from the website using the specific element like a hotel name, phone number, or address.

- Step 4: Cleaning and Validation

  Implement algorithms to eliminate duplicate entries and validate fields such as phone numbers, email addresses, and URLs. Apply relevance filters to eliminate irrelevant or redundant data, ensuring only high-quality, actionable information is retained.

- Step 5: Data Structuring

  Use Python libraries such as pandas to structure the cleaned data into a DataFrame.Export the DataFrame to Excel or CSV files, with proper formatting and headers for clarity.

- Step 6: Testing and Optimization

  Perform functional testing to verify data correctness and appropriateness. Optimize performance for large data sets to ensure scalability and reliability.

- Step 7: Deployment

  Pack the system as a standalone application or deploy it on cloud platforms to access the system remotely. Documentation or a help section to the end-user for understanding and using the system.

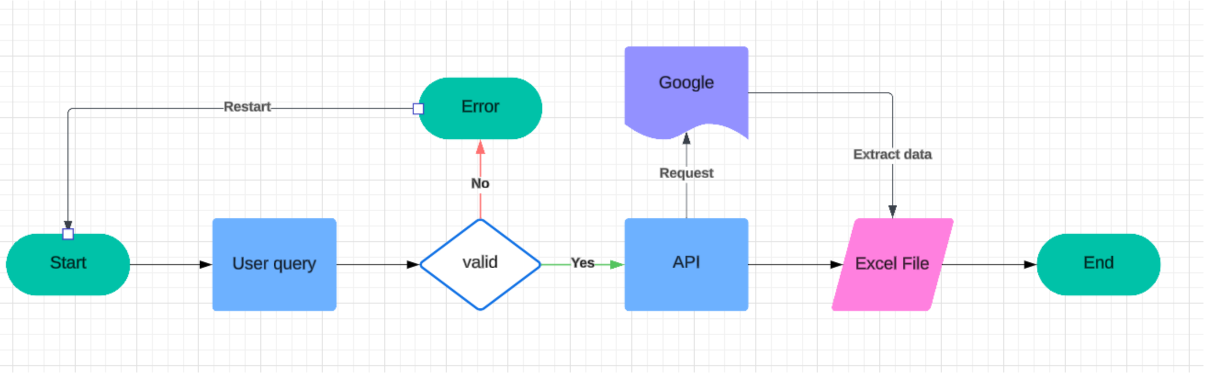


Figure 6.2: Workflow Diagram

6.3. Tools and Technologies

The following tools and technologies are used for the design and implementation of the system:

- RPA Tools:
  Selenium for browser automation and navigating search results.
- Web Scraping Tools:
  Beautiful Soup and Scrapy for parsing HTML and extracting relevant data.
- APIs:
  Rapid API, Google Places API, and similar services for accessing structured data.
- Data Processing Libraries:
  pandas for cleaning, filtering, and organizing data. OpenPyXL or XlsxWriter for exporting data into Excel or CSV formats.
- User Interface:
  Tkinter or PyQt for building a simple and interactive UI.
- Deployment Platforms:
  Desktop (Windows, macOS, Linux) for local applications. Cloud platforms (e.g., AWS, Google Cloud) for web-based deployment.

6.4 Key Design Features

- Modularity: Every module is self-contained and easy to maintain and update.
- Scalability: The system can process a huge number of queries and deal with various datasets without any performance degradation.
- Flexibility: Supports various domains (hotels, schools, restaurants, etc.) and user preferences.
- User-Friendliness: Simple UI to ensure accessibility to non-technical users.
- Ethical Compliance: API integration is prioritized to minimize the use of web scraping and maintain legality and ethics.

6.5 Challenges and Solutions

- Challenge 1: Handling dynamic content on web pages.
  Solution: Use Selenium with headless browsers to interact with JavaScript-driven websites.
- Challenge 2: Overcoming rate limits and captchas.
  Solution: Implement techniques like IP rotation and API usage to bypass these restrictions.
- Challenge 3: Ensuring data accuracy and relevance.

Solution: Advanced filtering and validation mechanism should be implemented during the cleaning phase.

CHAPTER-7
TIMELINE FOR EXECUTION OF PROJECT
(GANTT CHART)

7.1 Phase 1: Project Planning (October 2024, Week 1–Week 2)

Week 1:

- Scope, objectives, and deliverables of the project.
- Feasibility study for technical, financial, and resource-related constraints.

Week 2:

- Resources: team members, tools, and budget allocated for the project.
- Detailed roadmap: milestones, deadlines, and risk mitigation strategies.

7.2 Phase 2: System Design and Architecture (October 2024, Week 3–Week 4)

Week 3:

- Develop system flowcharts and architectural diagrams to describe the workflow of the project.
- Define the tools and technologies needed, RPA platforms, and APIs such as Rapid API.

Week 4:

- Develop mechanisms to implement unstructured and dynamic content handling such as JavaScript-driven websites
- Evolve an approach towards the tools needed to cope up with rate limits, captchas, and the bottlenecks arising from scalability issues.

7.3 Phase 3: Development and Setup (November 2024, Week 5–Week 9)

Week 5:

- Start developing the module to extract search results with RPA tools.
- Implement algorithms for cleaning and filtering raw data to be accurate and relevant.

Week 6:

- Mechanisms for exporting cleaned and structured data in the format of Excel and CSV

- Create an interface that accepts user queries like "hotels in Goa" and shows results

Week 7:

- APIs like Rapid API for better extraction of data along with ratings and descriptions.
- Check the compatibility between RPA scripts and API integration.

Week 8

- Carry out internal testing on each module to make sure that all works and solve some minor problems
- Implement query filters so that domain-based results can be shown in the interface.

Week 9

- Integrate all individual modules into one system.
- Improve performance of the system to accommodate huge data volumes and multiple types of queries.

7.4 Phase 4: Testing and Debugging (December 2024 to January 2025, Week 10–Week 14)

Week 10:

- Unit testing of all modules to check the correctness and efficiency of modules.
- Integration testing to test the proper interaction of all the components of the system.

Week 11:

- Test against real-world conditions, like handling dynamic content or large datasets.
- Analyze the system behaviour in rate limits or captchas and implement the appropriate fixes.

Week 12:

- Simulate user queries to evaluate the system’s usability and relevance of results.
- Address user feedback by resolving bugs and refining features.

Week 13:

- Optimize data filtering and cleaning algorithms to reduce duplicates and irrelevant data.
- Finalize the system’s core functionalities, ensuring all queries are processed efficiently.

Week 14:

- Verify the system’s scalability across different domains, such as hotels, schools, and restaurants.
- Perform stress testing on the system to ensure that it can sustain a high level of traffic

and big data sets.

7.5 Phase 5: Documentation (January 2025, Week 15–Week 16)

Week 15:

- User manuals describing how to operate the system and access results
- Technical documentation, which should include workflow, algorithms, and system architecture

Week 16:

- Final project report on objectives, methodology, and outcome
- Challenges experienced in the course of development and solutions adopted.

7.6 Phase 6: Deployment (February 2025, Week 17–Week 18)

Week 17:

- Deploy the system on target platforms with compatibility for different operating environments.
- Train users on how to operate the system with a focus on query input and interpretation of the output.

Week 18:

- Collect feedback from the first set of users to pinpoint any last-minute problems or areas of improvement.
- Fix deployment-specific issues, such as platform-specific bugs or usability issues.

7.7 Phase 7: Maintenance and Monitoring (February 2025 to March 2025, Week 19–Week 24)

Week 19:

- Monitor the performance of the system in real-world environments, focusing on data accuracy and processing speed.
- Gather user feedback on new requirements or potential improvements.

Week 20:

- Make system updates that reflect changing user needs and technological advancement.
- Tweak API integrations to fetch more accurate and diversified data.

Week 21:

- Periodic maintenance to ensure that the system is reliable and secure.
- Optimization of data handling mechanisms to enhance the performance of the entire system.

Week 22

- Features such as advanced filtering or domain-specific templates will be developed based on user feedback.
- Newly added features are tested to ensure that they do not affect existing functionalities.

Week 23

- Long-term performance metrics, such as scalability and user satisfaction, are analyzed.
- Remaining bugs or inefficiencies in the system are addressed.

Week 24

- Make the system stable and reliable for long-term use. Ensure that it is robust, user-friendly, and scalable.
- Give a final performance review and prepare for future iterations of the system.

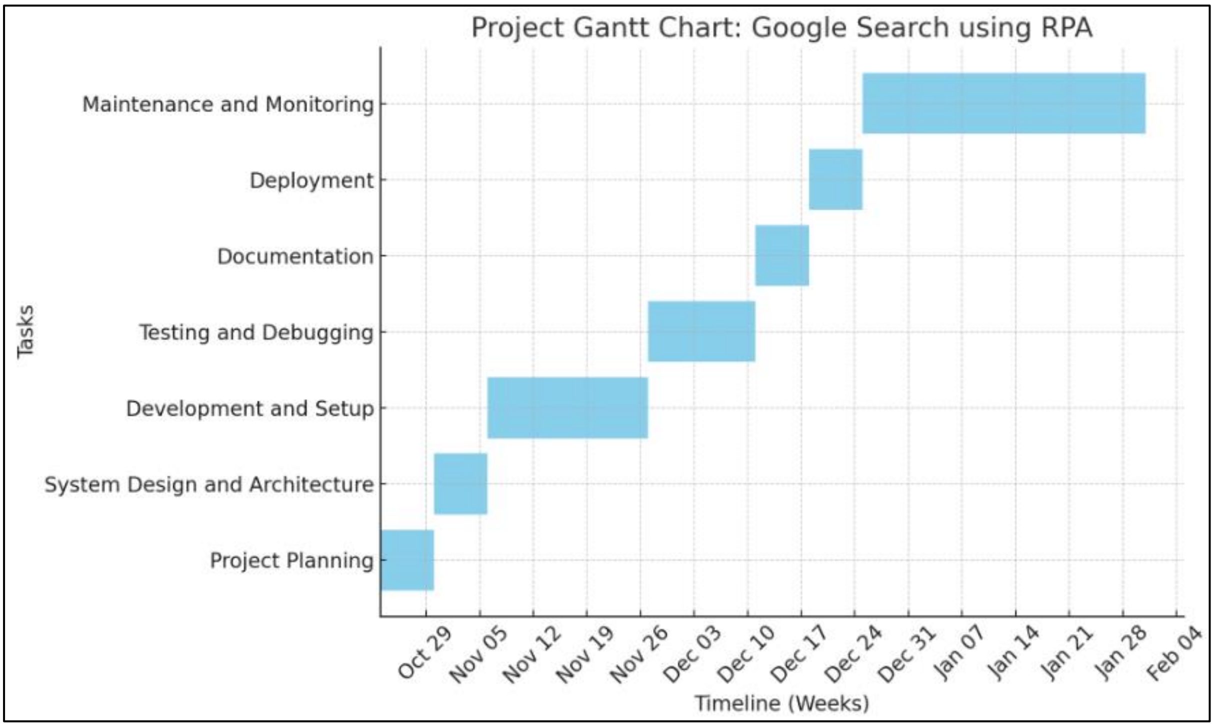


Figure 7.1 Gantt Chatt

CHAPTER-8
OUTCOMES

The system to be developed must provide a practical and user-friendly application that fetches data automatically from Google search results, verifies data accuracy and relevance, and outputs them in formatted forms like Excel or CSV. Below are the expected results of the project classified under functional, technical, and general impacts.

8.1 Functional Results

- **Automatic Data Retrieval:**
  Automates relevant data extraction including hotels' names, contact numbers, email addresses, addresses, and links, Structured information will be generated for queries; for example, "hotels in Goa". No human effort is required while getting structured information.
- **Clean and usable data**
  The system will have to provide the results in clean format, that would be easily accessible for analysis purpose, either through Excel or CSV, every output row will be an instance, which contains columns like name, address, and contact details.
- **Efficient Query Handling:**
  System accepts different types of queries from diverse domains such as hotels, schools, restaurants, The results would come out to be much quicker than manual searching

8.2 Technical Outcomes

- **RPA API Integration:**
  - ➢ The project will show the integration of Robotic Process Automation (RPA) tools like Selenium with APIs like Rapid API or Google Places API.
  - ➢ This ensures that data extraction is accurate and that there is a minimum amount of manual interaction required with web pages.

- **Advanced Data Cleaning and Filtering:**

Algorithms implemented to remove duplicates, validate contact information, and ensure relevance.

The system will filter out advertisements, irrelevant data, and other non-essential elements from search results.

- **Handling Dynamic Content:**
  - ➢ The system will demonstrate the ability to interact with dynamic, JavaScript-driven web pages using tools like Selenium.
  - ➢ Dynamic content extraction ensures comprehensive data retrieval even from modern websites.

- **Scalable and Flexible Design:**

Scalability to any size of the dataset and cross-domain adaptability will ensure it performs consistently whether the query type is large or not.

- **Ethical Compliance:**

This makes sure that due to API utilization instead of web scraping, there's a reduced probability of breaching a policy or data protection act.

8.3 User-Focused Results

- **Convenient Data Extraction:**

The system will give a user-friendly interface for non-technical users to input queries and retrieve data effortlessly.

Less steps and intuitive design will increase accessibility and usability.

- **Time and Cost Efficiency:**

The system will reduce the time and cost incurred in manual data collection by automating repetitive tasks.

Small businesses, researchers, and individuals will have streamlined workflows and faster decision-making.

- **Better Decision-Making:**

Structured and reliable data will enable the users to take decisions, either for business planning, marketing, or research purposes.

The output can be used further for analysis or integration into other systems.

8.4 Wider Implications

- **Interdomain Utility:**

  The system, though implemented for hotels, can be applied to other domains like schools, restaurants, or local businesses as well, making it versatile.

- **Facilitating Ethical Automation:**

  By incorporating APIs and following the ethical standards, the project showcases how automation can be done responsibly and sets a precedent for future automation projects.

- **Contribution to Research and Development:**

  The design and methodology of the system will give insights into filling the gaps in the current data extraction methods, contributing to advancements in automation and API utilization.

- **Educational Value:**

  This is to be an all-learning platform for individuals and organizations interested in RPA, web scraping, and API integration.

8.5 Quantifiable Results

- **Accuracy**: It provides high-quality data, cleaned up and validated without much duplication and errors.

  Efficiency: Reduce data retrieval by at least 90% with a comparison made from the existing process.
- **Scalability**: it can effectively run up to thousands of queries with large amounts of data
- **Ease of Use:** Simple workflow where little to no technical skill in terms of operating this program.

CHAPTER-9

RESULTS AND DISCUSSIONS

9.1 Data Overview

The output contains the data pertaining to hotels at Hyderabad in which detailed information including ID, name and description with respective ranking positions of each along with its rating has been obtained through a booking API on an interaction through RPA BOT.

Hotel ID	Name	Description	Ranking Position	Rating
10086624	Hotel silver Cle	Hotel silver	1	9.8
13194508	Blossoms Room the plaza	Blossom	2	9.5
13088080	FabHotel	FabHotel	3	8.9

Table 9.1: Sample Data

9.2 Main Conclusion

- **Hotel Ratings:**
  - The highest rank is held by "Blossoms Room The Plaza- Free Lunch Or Dinner" with 1 ranking and a rating score of 9.5
  - The second-best ranked hotel is "FabHotel Prime Srishti Inn - HITECH CITY" with the ranking position as 2 with a rating of 9

Hotel ID	Name	Description	Ranking Position	Rating
13194950	Blossoms Room The Plaza	Blossom	1	9.5

13088808	FabHotel Prime Srishti	FabHotel	2	9
10958635	FabHotel Hitech	FabHotel	3	8.9

Table9.2: Top-Ranked Hotels

- **Rating Distribution:**
  - Most rated hotel- "Blossoms Room the Plaza- Free Lunch or Dinner" with the rating score being 9.5.
  - The lowest-rated hotel is "NK Homes - Urban Serviced Apartments" with a rating of 7.9.

Rating	Number of Hotels
9.5	2
9	4
8.5	5
8.1	7
7.9	8

Table 9.2: Rating Distribution

- **Hotel Descriptions:**

  Descriptions are a short description of each hotel, which can be helpful for guests to make the right choice.

- **Ranking Positions:**

  Ranking positions are the relative popularity or quality of the hotels based on user reviews and other factors.

9.3 Detailed Analysis

- **Hotel IDs:** Unique identifiers for each hotel, useful for further data processing and integration with other systems.

- **Names:** Detailed and descriptive names for the hotels to make easy identification.
- **Descriptions:** Short descriptions which outline the main feature or offering of each hotel.
- **Ranking Positions:** Numerical values that show the relative ranking of the hotels
- **Rating:** Users' ratings, which give the overall satisfaction and quality of the hotels.

9.4 Discussion

Several benefits have been realized by using RPA.

- **Efficiency:** The RPA bot efficiently retrieved and processed data from the Booking.com API, thus avoiding manual data entry and search.
- **Accuracy:** Automation ensured that the data retrieved was accurate and consistent, thereby reducing human errors.
- **Speed:** The process was completed very fast, which meant that the data retrieval and analysis were in real-time or near-real-time.

Metric	Value	Unit
Execution Time	5 sec	Second
Data Retrieval Speed	2 sec	Second
Number of API Calls	10	Calls
Average Response Time	0.5 sec	Seconds

Table 9.4: Performance Metrics

9.5 Challenges and Limitations

- **API Rate Limits:** The Booking.com API may be throttled which means that it can affect the speed of data retrieval and volume.
- **Partial Data:** The data retrieved is only a result of what the API provides. Some hotels may have partial data/missing information
- **Robust Error Handling:** There is a need to have robust error handling in case of

network errors, API being down, or change of API structure.

9.6 Future Improvements

- **Enhanced Data Analysis:** Include more data sources or APIs to offer more comprehensive information about each hotel, such as reviews, amenities, and pricing.
- **User Interface:** Develop a user-friendly interface for users to input their preferences and receive personalized hotel recommendations.
- **Real-Time Updates:** Implement real-time updates to ensure that the data is always current and accurate.
- **Machine Learning:** Use machine learning algorithms to predict the rating of a hotel or recommend hotels as per user preferences with historical data.

CHAPTER-10
CONCLUSION

The project aimed at automating the retrieval and processing of hotel data from the Booking.com API using Robotic Process Automation (RPA). The successful implementation of this project has yielded valuable insights into the hotel offerings in Hyderabad, providing a comprehensive list of 21 hotels along with their respective details such as Hotel ID, Name, Description, Ranking Position, and Rating. This information can be very useful for travelers, hoteliers, and travel agencies in making informed decisions and understanding the competitive landscape of the hotel market in Hyderabad.

One of the key findings from the project is the identification of top-ranked hotels. "Blossoms Room The Plaza- Free Lunch Or Dinner" emerged as the highest-ranked hotel with a ranking position of 1 and an impressive rating of 9.5. This hotel owns the number one ranking but also has the attractive feature of offering free lunch or dinner, which might be a substantial selling point for potential guests. Ranking close to them is "FabHotel Prime Srishti Inn - HITECH CITY". Rating 9 and ranking in the position 2. Rating indicates that most guests are fully satisfied with what the hotels offered to them as regards quality service and amenities during their stay, which sets it as first-class accommodation services.

The rating distribution of the listed hotels ranges from 7.3 to 9.5, with most of the hotels achieving ratings above 8. This indicates a general high standard of accommodations in Hyderabad, which is in line with the competitive nature of the hotel market. The descriptions given for each hotel provide brief yet valuable insights into their unique features and offerings, which may help potential guests in making a decision. For example, "Free Lunch Or Dinner" for "Blossoms Room The Plaza" and detailed location information for hotels such as "FabHotel Golden HITECH - HITECH CITY" are vital information that might influence a guest's choice.

This project has already shown several benefits through the use of RPA: efficiency, accuracy, and speed. The RPA bot was able to fetch and process data from the Booking.com API with a minimal amount of manual data entry and search. The automation of this process guaranteed that the retrieved data was correct and consistent, with minimal chances of human errors. The fast pace at which the process was completed allowed for real-time or near-real-time data retrieval and analysis, which made the system very responsive to user queries.

However, there were also challenges and limitations with the project. The Booking.com API may also have rate limits, which may affect the speed and volume of data retrieval. Also, the retrieved data is limited to what is available through the API, and some hotels may have incomplete or missing information. Robust error handling is also necessary to manage issues such as network errors, API downtime, or changes in the API structure. These pose a call for constant monitoring and adjustments to the RPA system to guarantee its credibility and efficiency.

REFERENCES

[1] S. Aguirre and A. Rodriguez, "Automation in the travel industry: How RPA is transforming business operations," *Journal of Business Process Management*, vol. 23, no. 3, pp. 345–356, 2017.

[2] L. P. Willcocks, M. C. Lacity, and A. Craig, "The IT function and Robotic Process Automation," *The Outsourcing Unit Working Research Paper Series*, vol. 15, no. 5, pp. 1–25, 2015.

[3] M. Singh and P. Jain, "Web scraping and API integration: Simplifying data gathering," *International Journal of Data Science*, vol. 6, no. 2, pp. 45–53, 2018.

[4] K. Sharma, R. Singh, and P. Jain, "Automating data processing using Python-based tools," *International Journal of Computer Applications*, vol. 178, no. 32, pp. 12–18, 2019.

[5] D. Zeng, Z. Lu, and B. Guo, "Real-time data extraction in the hospitality industry using APIs," *Tourism Management Perspectives*, vol. 38, p. 100821, 2021.

[6] J. Brock and A. Markou, "Robotic Process Automation: A disruptive technology," *Journal of Business Research*, vol. 125, pp. 476–487, 2021.

[7] A. Gupta and P. R. Saini, "Application of APIs in automating data extraction processes," *International Journal of Computer Applications*, vol. 182, no. 22, pp. 20–25, 2020.

[8] R. K. Jain, "The role of Python in modern automation systems," *IEEE International Conference on Automation Science and Engineering*, pp. 112–118, 2018.

[9] N. Kumar and S. Goel, "Automation of data collection in the hospitality sector using Python and APIs," *Journal of Emerging Technologies and Innovative Research*, vol. 6, no. 5, pp. 95–102, 2019.

[10] T. Lin and H. Chen, "APIs and their role in scalable data integration," *IEEE Transactions on Services Computing*, vol. 12, no. 1, pp. 45–56, 2019.

[11] P. Raj and R. Kumar, "A comprehensive study on the use of Robotic Process Automation in various industries," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 8, pp. 350–356, 2019.

[12] G. Behera and M. Dash, "Data extraction and visualization using Python," *International Journal of Innovative Research in Computer Science & Technology*, vol. 6, no. 1, pp. 42–47, 2018.

[13] Rapid API, "Booking.com API Documentation," *Rapid API*

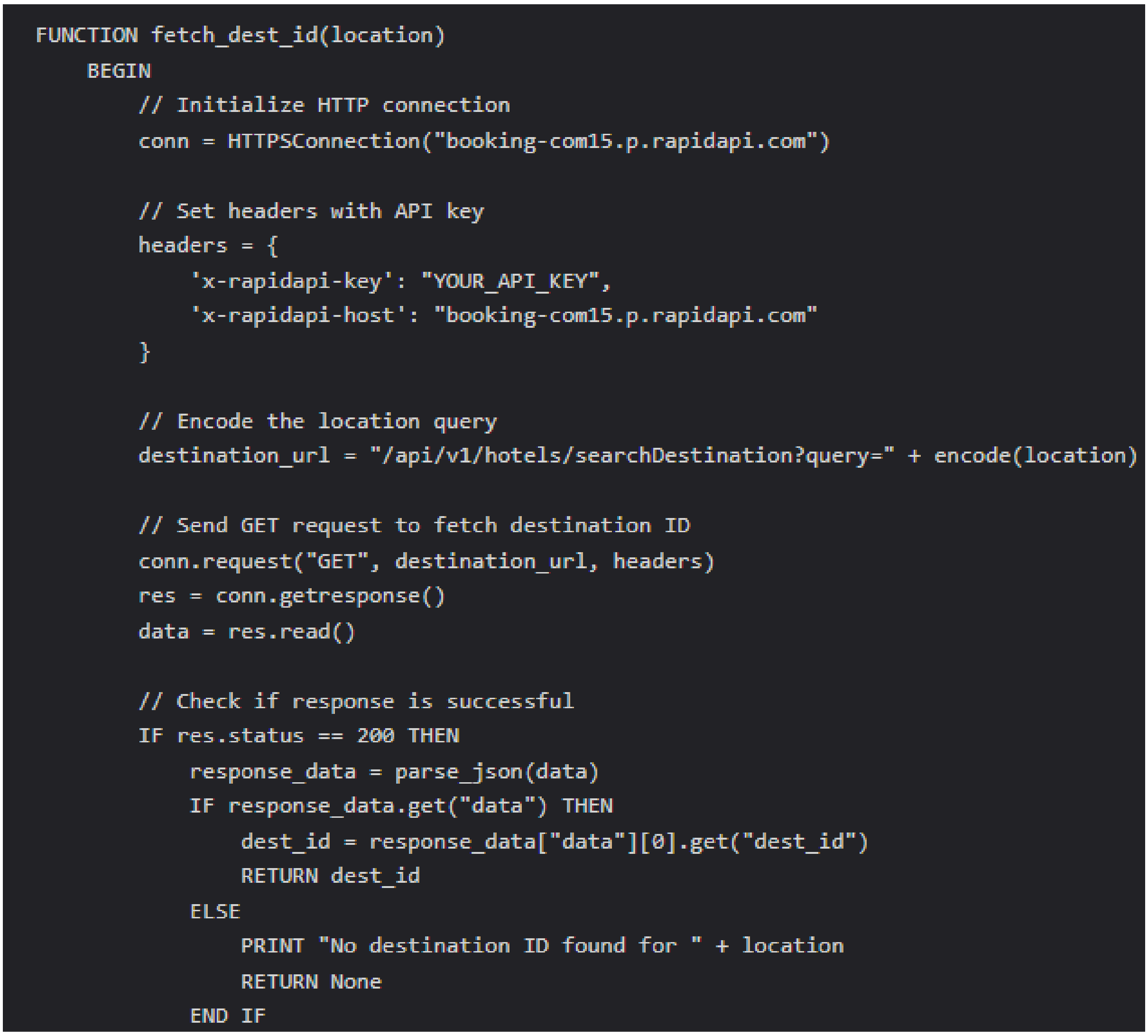
[14] W. Fan, "Data scraping and analysis for business decision-making," *International Conference on Data Mining and Machine Learning*, pp. 65–72, 2018.

[15]B. Hall and L. Kretschmer, "Digital transformation with automation and APIs," *MIT Sloan Management Review*, vol. 61, no. 2, pp. 42–49, 2020.

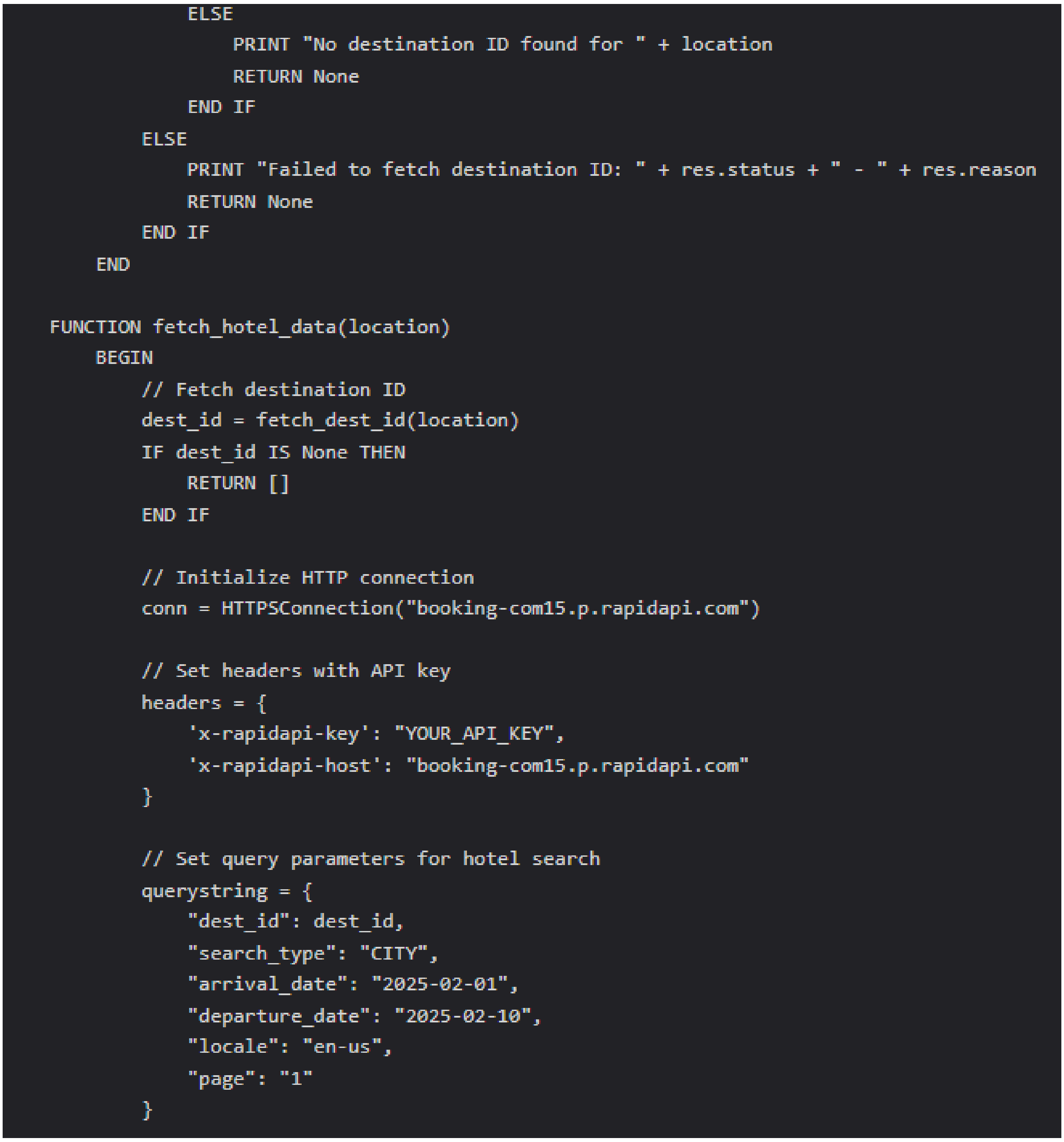
APPENDIX-A

PSUEDOCODE

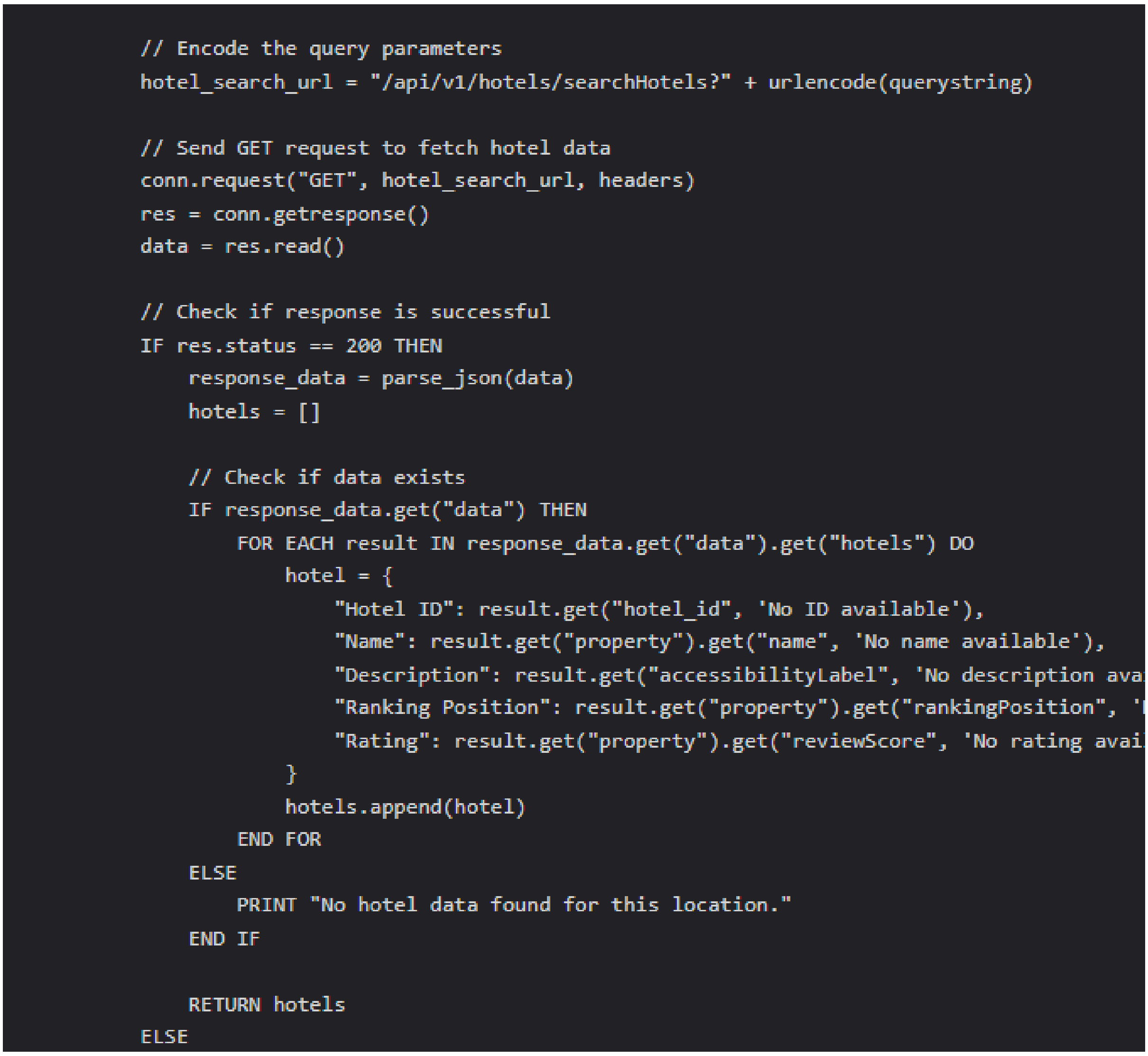
BEGIN



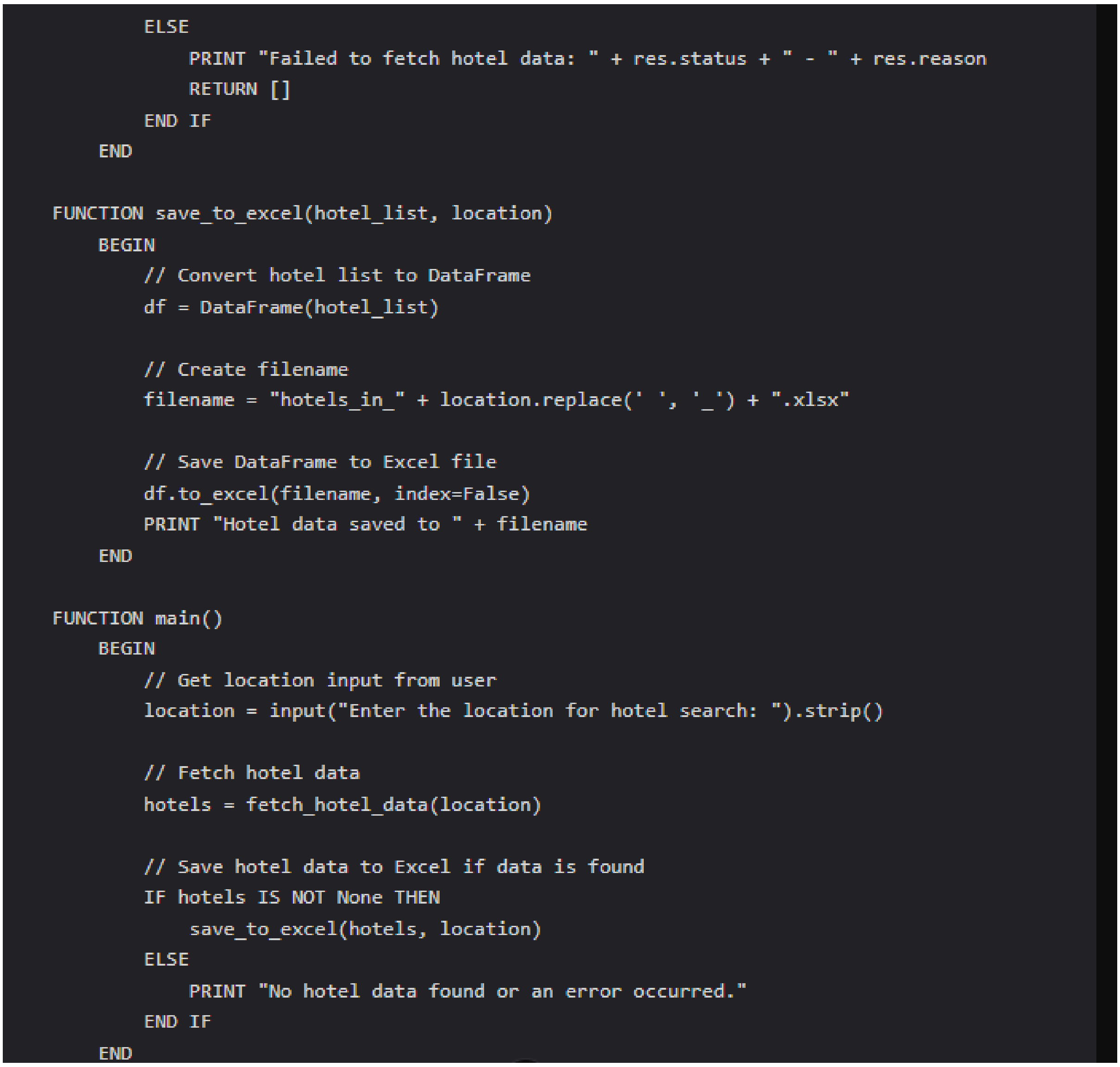
```
FUNCTION fetch_dest_id(location)
    BEGIN
        // Initialize HTTP connection
        conn = HTTPSConnection("booking-com15.p.rapidapi.com")

        // Set headers with API key
        headers = {
            'x-rapidapi-key': "YOUR_API_KEY",
            'x-rapidapi-host': "booking-com15.p.rapidapi.com"
        }

        // Encode the location query
        destination_url = "/api/v1/hotels/searchDestination?query=" + encode(location)

        // Send GET request to fetch destination ID
        conn.request("GET", destination_url, headers)
        res = conn.getresponse()
        data = res.read()

        // Check if response is successful
        IF res.status == 200 THEN
            response_data = parse_json(data)
            IF response_data.get("data") THEN
                dest_id = response_data["data"][0].get("dest_id")
                RETURN dest_id
            ELSE
                PRINT "No destination ID found for " + location
                RETURN None
            END IF
```



```
            ELSE
                PRINT "No destination ID found for " + location
                RETURN None
            END IF
        ELSE
            PRINT "Failed to fetch destination ID: " + res.status + " - " + res.reason
            RETURN None
        END IF
    END

FUNCTION fetch_hotel_data(location)
    BEGIN
        // Fetch destination ID
        dest_id = fetch_dest_id(location)
        IF dest_id IS None THEN
            RETURN []
        END IF

        // Initialize HTTP connection
        conn = HTTPSConnection("booking-com15.p.rapidapi.com")

        // Set headers with API key
        headers = {
            'x-rapidapi-key': "YOUR_API_KEY",
            'x-rapidapi-host': "booking-com15.p.rapidapi.com"
        }

        // Set query parameters for hotel search
        querystring = {
            "dest_id": dest_id,
            "search_type": "CITY",
            "arrival_date": "2025-02-01",
            "departure_date": "2025-02-10",
            "locale": "en-us",
            "page": "1"
        }
```



```
// Encode the query parameters
hotel_search_url = "/api/v1/hotels/searchHotels?" + urlencode(querystring)

// Send GET request to fetch hotel data
conn.request("GET", hotel_search_url, headers)
res = conn.getresponse()
data = res.read()

// Check if response is successful
IF res.status == 200 THEN
    response_data = parse_json(data)
    hotels = []

    // Check if data exists
    IF response_data.get("data") THEN
        FOR EACH result IN response_data.get("data").get("hotels") DO
            hotel = {
                "Hotel ID": result.get("hotel_id", 'No ID available'),
                "Name": result.get("property").get("name", 'No name available'),
                "Description": result.get("accessibilityLabel", 'No description ava
                "Ranking Position": result.get("property").get("rankingPosition", '
                "Rating": result.get("property").get("reviewScore", 'No rating avai
            }
            hotels.append(hotel)
        END FOR
    ELSE
        PRINT "No hotel data found for this location."
    END IF

    RETURN hotels
ELSE
```



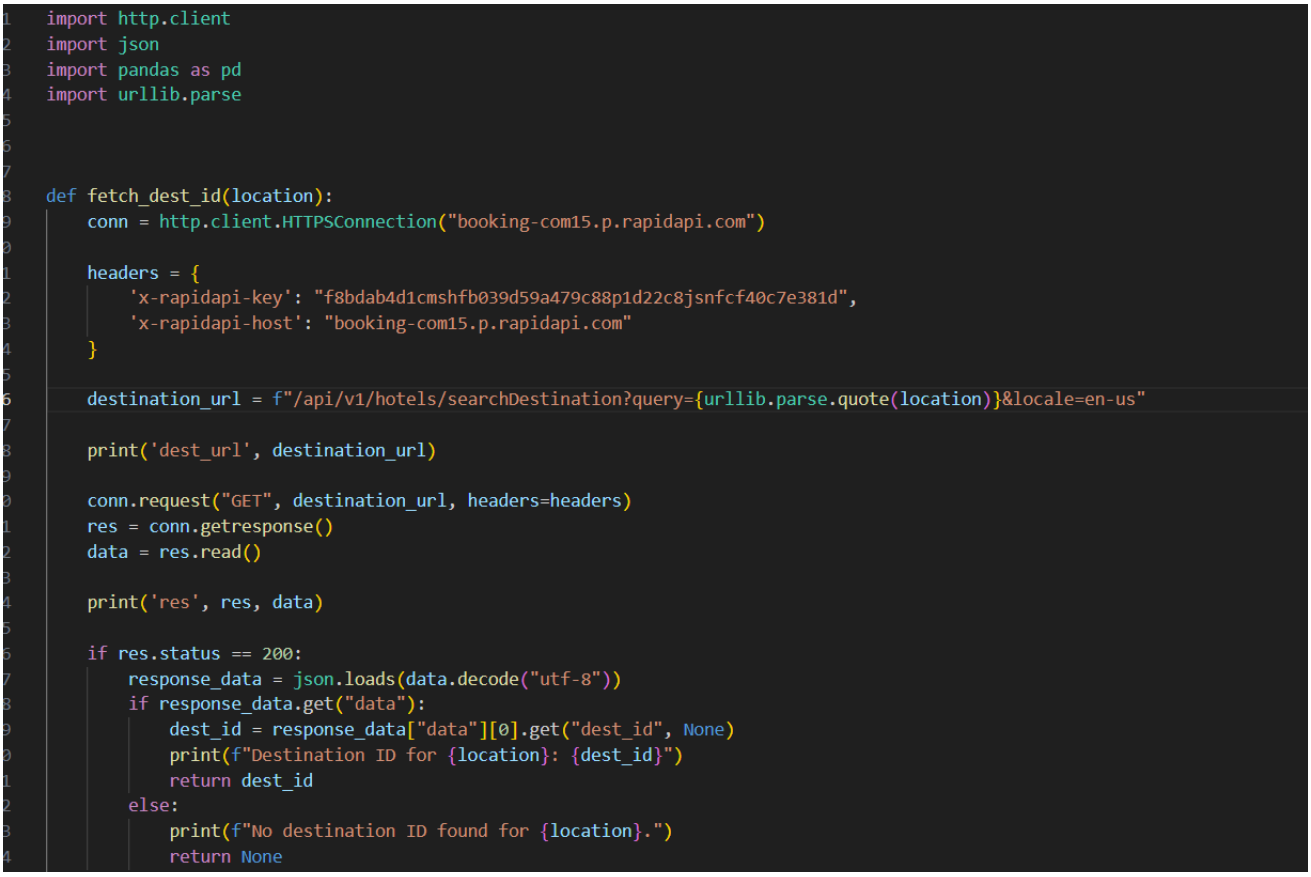
```
            ELSE
                PRINT "Failed to fetch hotel data: " + res.status + " - " + res.reason
                RETURN []
            END IF
        END

    FUNCTION save_to_excel(hotel_list, location)
        BEGIN
            // Convert hotel list to DataFrame
            df = DataFrame(hotel_list)

            // Create filename
            filename = "hotels_in_" + location.replace(' ', '_') + ".xlsx"

            // Save DataFrame to Excel file
            df.to_excel(filename, index=False)
            PRINT "Hotel data saved to " + filename
        END

    FUNCTION main()
        BEGIN
            // Get location input from user
            location = input("Enter the location for hotel search: ").strip()

            // Fetch hotel data
            hotels = fetch_hotel_data(location)

            // Save hotel data to Excel if data is found
            IF hotels IS NOT None THEN
                save_to_excel(hotels, location)
            ELSE
                PRINT "No hotel data found or an error occurred."
            END IF
        END
```

END

APPENDIX-B

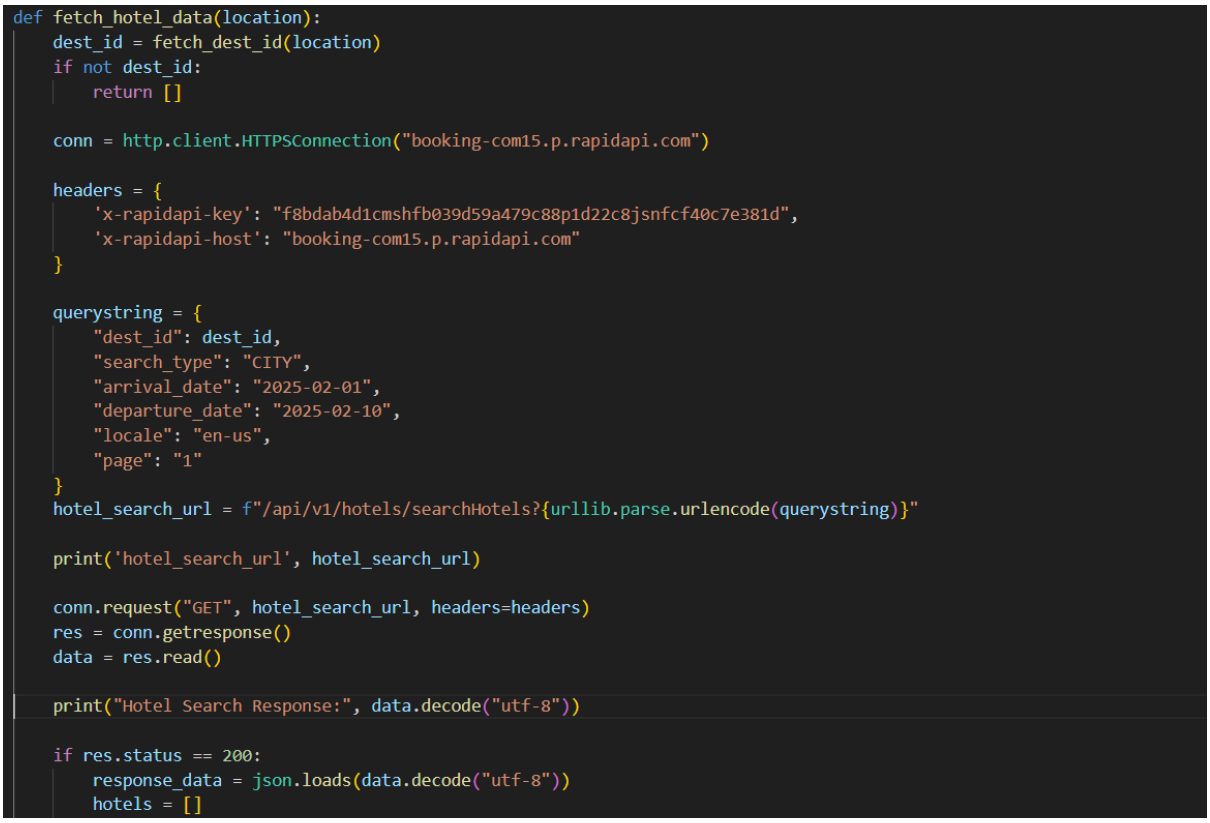
SCREENSHOTS

1. Fetch dest_id based on the user input location



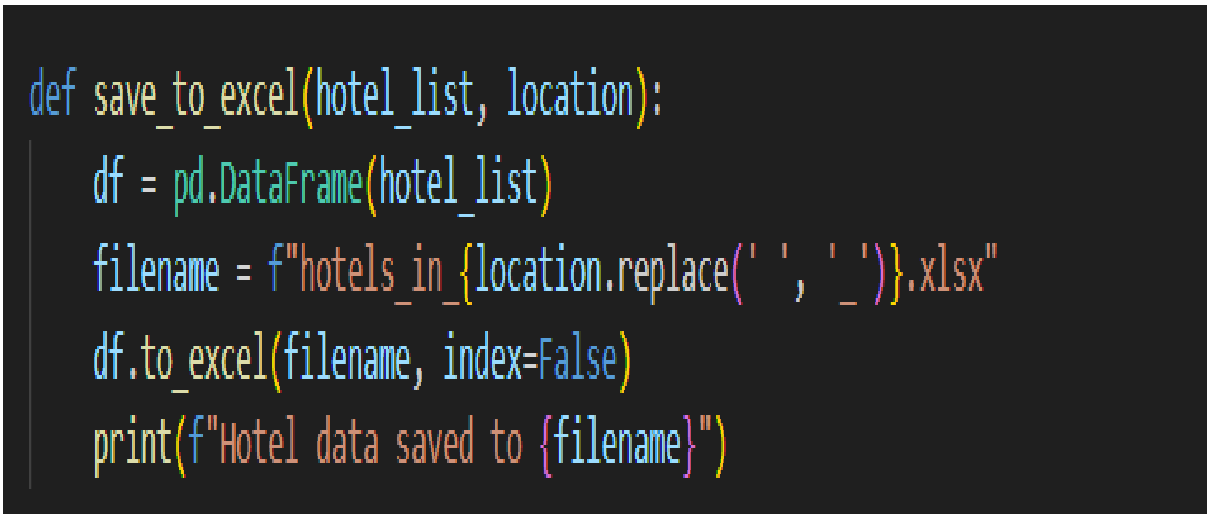
```python
import http.client
import json
import pandas as pd
import urllib.parse



def fetch_dest_id(location):
    conn = http.client.HTTPSConnection("booking-com15.p.rapidapi.com")

    headers = {
        'x-rapidapi-key': "f8bdab4d1cmshfb039d59a479c88p1d22c8jsnfcf40c7e381d",
        'x-rapidapi-host': "booking-com15.p.rapidapi.com"
    }

    destination_url = f"/api/v1/hotels/searchDestination?query={urllib.parse.quote(location)}&locale=en-us"

    print('dest_url', destination_url)

    conn.request("GET", destination_url, headers=headers)
    res = conn.getresponse()
    data = res.read()

    print('res', res, data)

    if res.status == 200:
        response_data = json.loads(data.decode("utf-8"))
        if response_data.get("data"):
            dest_id = response_data["data"][0].get("dest_id", None)
            print(f"Destination ID for {location}: {dest_id}")
            return dest_id
        else:
            print(f"No destination ID found for {location}.")
            return None
```

2.Use dest_id to fetch hotel data (No check-in/checkout dates)



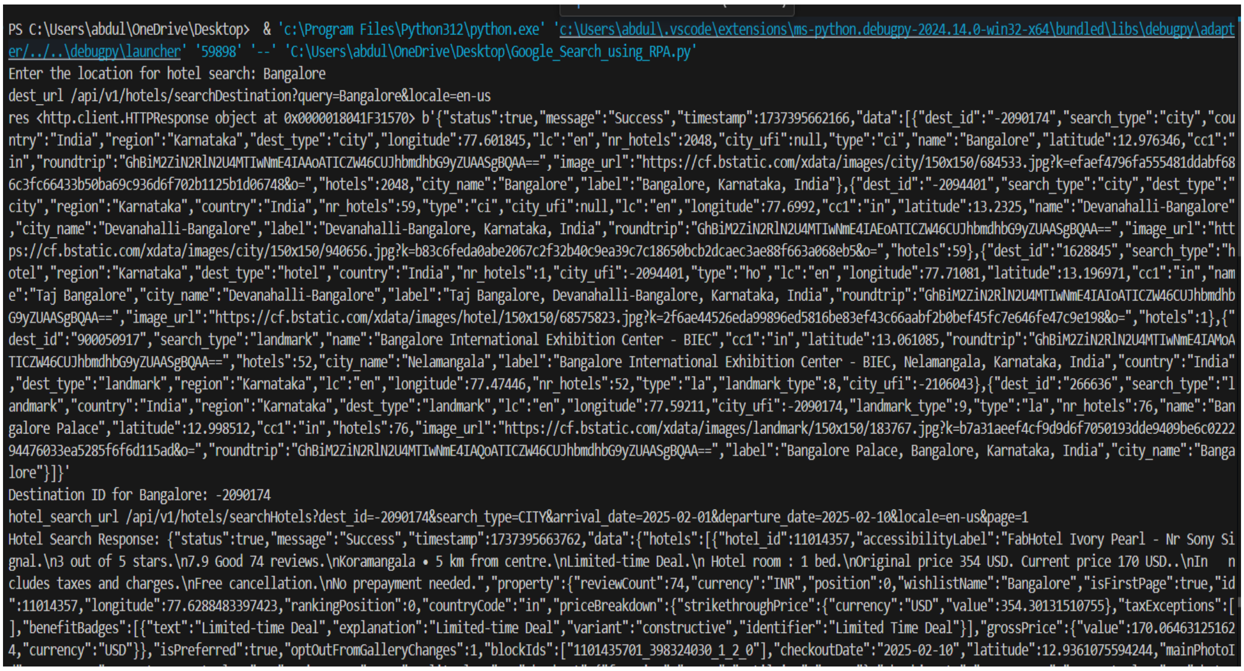
```
def fetch_hotel_data(location):
    dest_id = fetch_dest_id(location)
    if not dest_id:
        return []

    conn = http.client.HTTPSConnection("booking-com15.p.rapidapi.com")

    headers = {
        'x-rapidapi-key': "f8bdab4d1cmshfb039d59a479c88p1d22c8jsnfcf40c7e381d",
        'x-rapidapi-host': "booking-com15.p.rapidapi.com"
    }

    querystring = {
        "dest_id": dest_id,
        "search_type": "CITY",
        "arrival_date": "2025-02-01",
        "departure_date": "2025-02-10",
        "locale": "en-us",
        "page": "1"
    }
    hotel_search_url = f"/api/v1/hotels/searchHotels?{urllib.parse.urlencode(querystring)}"

    print('hotel_search_url', hotel_search_url)

    conn.request("GET", hotel_search_url, headers=headers)
    res = conn.getresponse()
    data = res.read()

    print("Hotel Search Response:", data.decode("utf-8"))

    if res.status == 200:
        response_data = json.loads(data.decode("utf-8"))
        hotels = []
```

1. Save hotel data to Excel



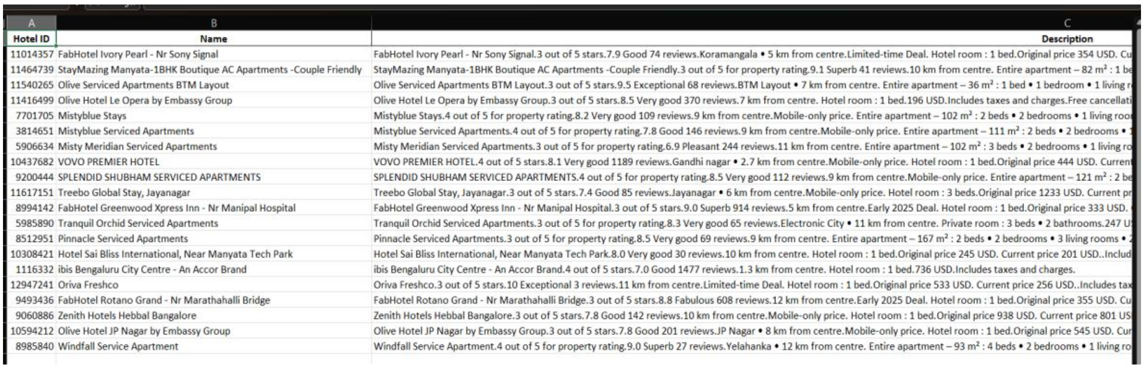
```python
def save_to_excel(hotel_list, location):
    df = pd.DataFrame(hotel_list)
    filename = f"hotels_in_{location.replace(' ', '_')}.xlsx"
    df.to_excel(filename, index=False)
    print(f"Hotel data saved to {filename}")
```

2. User Input

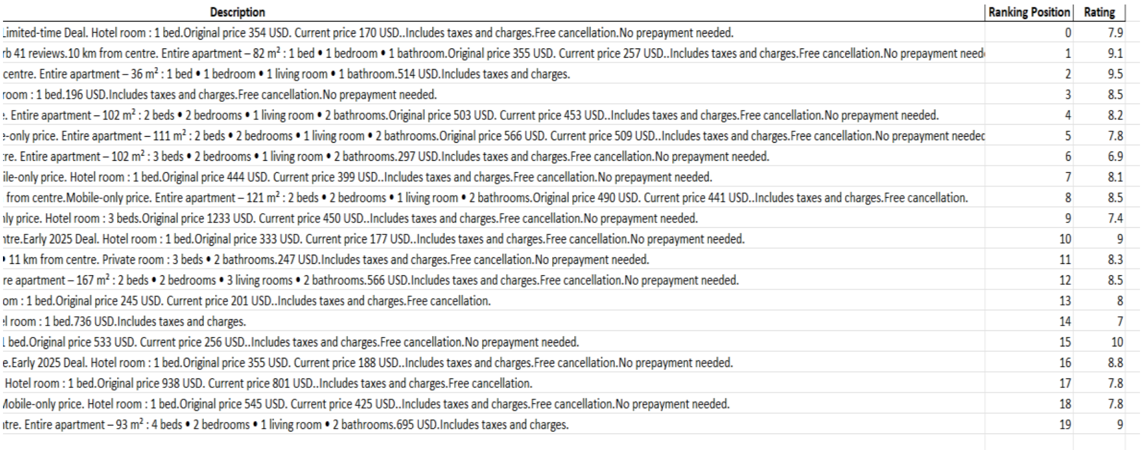


```
PS C:\Users\abdul\OneDrive\Desktop>  & 'c:\Program Files\Python312\python.exe' 'c:\Users\abdul\.vscode\extensions\ms-python.debugpy-2024.14.0-win32-x64\bundled\libs\debugpy\adapt
er/../..\debugpy\launcher' '59898' '--' 'C:\Users\abdul\OneDrive\Desktop\Google_Search_using_RPA.py'
Enter the location for hotel search: Bangalore
dest_url /api/v1/hotels/searchDestination?query=Bangalore&locale=en-us
res <http.client.HTTPResponse object at 0x0000018041F31570> b'{"status":true,"message":"Success","timestamp":1737395662166,"data":[{"dest_id":"-2090174","search_type":"city","cou
ntry":"India","region":"Karnataka","dest_type":"city","longitude":77.601845,"lc":"en","nr_hotels":2048,"city_ufi":null,"type":"ci","name":"Bangalore","latitude":12.976346,"cc1":"
in","roundtrip":"GhBiM2ZiN2RlN2U4MTIwNmE4IAAoATICZW46CUJhbmdhbG9yZUAASgBQAA==","image_url":"https://cf.bstatic.com/xdata/images/city/150x150/684533.jpg?k=efaef4796fa555481ddabf68
6c3fc66433b50ba69c936d6f702b1125b1d06748&o=","hotels":2048,"city_name":"Bangalore","label":"Bangalore, Karnataka, India"},{"dest_id":"-2094401","search_type":"city","dest_type":"
city","region":"Karnataka","country":"India","nr_hotels":59,"type":"ci","city_ufi":null,"lc":"en","longitude":77.6992,"cc1":"in","latitude":13.2325,"name":"Devanahalli-Bangalore"
,"city_name":"Devanahalli-Bangalore","label":"Devanahalli-Bangalore, Karnataka, India","roundtrip":"GhBiM2ZiN2RlN2U4MTIwNmE4IAEoATICZW46CUJhbmdhbG9yZUAASgBQAA==","image_url":"htt
ps://cf.bstatic.com/xdata/images/city/150x150/940656.jpg?k=b83c6feda0abe2067c2f32b40c9ea39c7c18650bcb2dcaec3ae88f663a068eb5&o=","hotels":59},{"dest_id":"1628845","search_type":"h
otel","region":"Karnataka","dest_type":"hotel","country":"India","nr_hotels":1,"city_ufi":-2094401,"type":"ho","lc":"en","longitude":77.71081,"latitude":13.196971,"cc1":"in","nam
e":"Taj Bangalore","city_name":"Devanahalli-Bangalore","label":"Taj Bangalore, Devanahalli-Bangalore, Karnataka, India","roundtrip":"GhBiM2ZiN2RlN2U4MTIwNmE4IAIoATICZW46CUJhbmdhb
G9yZUAASgBQAA==","image_url":"https://cf.bstatic.com/xdata/images/hotel/150x150/68575823.jpg?k=2f6ae44526eda99896ed5816be83ef43c66aabf2b0bef45fc7e646fe47c9e198&o=","hotels":1},{"
dest_id":"900050917","search_type":"landmark","name":"Bangalore International Exhibition Center - BIEC","cc1":"in","latitude":13.061085,"roundtrip":"GhBiM2ZiN2RlN2U4MTIwNmE4IAMoA
TICZW46CUJhbmdhbG9yZUAASgBQAA==","hotels":52,"city_name":"Nelamangala","label":"Bangalore International Exhibition Center - BIEC, Nelamangala, Karnataka, India","country":"India"
,"dest_type":"landmark","region":"Karnataka","lc":"en","longitude":77.47446,"nr_hotels":52,"type":"la","landmark_type":8,"city_ufi":-2106043},{"dest_id":"266636","search_type":"l
andmark","country":"India","region":"Karnataka","dest_type":"landmark","lc":"en","longitude":77.59211,"city_ufi":-2090174,"landmark_type":9,"type":"la","nr_hotels":76,"name":"Ban
galore Palace","latitude":12.998512,"cc1":"in","hotels":76,"image_url":"https://cf.bstatic.com/xdata/images/landmark/150x150/183767.jpg?k=b7a31aeef4cf9d9d6f7050193dde9409be6c0222
94476033ea5285f6f6d115ad&o=","roundtrip":"GhBiM2ZiN2RlN2U4MTIwNmE4IAQoATICZW46CUJhbmdhbG9yZUAASgBQAA==","label":"Bangalore Palace, Bangalore, Karnataka, India","city_name":"Banga
lore"}]}'
Destination ID for Bangalore: -2090174
hotel_search_url /api/v1/hotels/searchHotels?dest_id=-2090174&search_type=CITY&arrival_date=2025-02-01&departure_date=2025-02-10&locale=en-us&page=1
Hotel Search Response: {"status":true,"message":"Success","timestamp":1737395663762,"data":{"hotels":[{"hotel_id":11014357,"accessibilityLabel":"FabHotel Ivory Pearl - Nr Sony Si
gnal.\n3 out of 5 stars.\n7.9 Good 74 reviews.\nKoramangala • 5 km from centre.\nLimited-time Deal.\n Hotel room : 1 bed.\nOriginal price 354 USD. Current price 170 USD..\nIn   n
cludes taxes and charges.\nFree cancellation.\nNo prepayment needed.","property":{"reviewCount":74,"currency":"INR","position":0,"wishlistName":"Bangalore","isFirstPage":true,"id
":11014357,"longitude":77.6288483397423,"rankingPosition":0,"countryCode":"in","priceBreakdown":{"strikethroughPrice":{"currency":"USD","value":354.30131510755},"taxExceptions":[
],"benefitBadges":[{"text":"Limited-time Deal","explanation":"Limited-time Deal","variant":"constructive","identifier":"Limited Time Deal"}],"grossPrice":{"value":170.06463125162
4,"currency":"USD"}},"isPreferred":true,"optOutFromGalleryChanges":1,"blockIds":["1101435701_398324030_1_2_0"],"checkoutDate":"2025-02-10","latitude":12.9361075594244,"mainPhotoI
```

3. Result



Hotel ID	Name	Description
11014357	FabHotel Ivory Pearl - Nr Sony Signal	FabHotel Ivory Pearl - Nr Sony Signal.3 out of 5 stars.7.9 Good 74 reviews.Koramangala • 5 km from centre.Limited-time Deal. Hotel room : 1 bed.Original price 354 USD. Cu
11464739	StayMazing Manyata-1BHK Boutique AC Apartments -Couple Friendly	StayMazing Manyata-1BHK Boutique AC Apartments -Couple Friendly.3 out of 5 for property rating.9.1 Superb 41 reviews.10 km from centre. Entire apartment – 82 m^2 : 1 be
11540265	Olive Serviced Apartments BTM Layout	Olive Serviced Apartments BTM Layout.3 out of 5 stars.9.5 Exceptional 68 reviews.BTM Layout • 7 km from centre. Entire apartment – 36 m^2 : 1 bed • 1 bedroom • 1 living r
11416499	Olive Hotel Le Opera by Embassy Group	Olive Hotel Le Opera by Embassy Group.3 out of 5 stars.8.5 Very good 370 reviews.7 km from centre. Hotel room : 1 bed.196 USD.Includes taxes and charges.Free cancellati
7701705	Mistyblue Stays	Mistyblue Stays.4 out of 5 for property rating.8.2 Very good 109 reviews.9 km from centre.Mobile-only price. Entire apartment – 102 m^2 : 2 beds • 2 bedrooms • 1 living roo
3814651	Mistyblue Serviced Apartments	Mistyblue Serviced Apartments.4 out of 5 for property rating.7.8 Good 146 reviews.9 km from centre.Mobile-only price. Entire apartment – 111 m^2 : 2 beds • 2 bedrooms •
5906634	Misty Meridian Serviced Apartments	Misty Meridian Serviced Apartments.3 out of 5 for property rating.6.9 Pleasant 244 reviews.11 km from centre. Entire apartment – 102 m^2 : 3 beds • 2 bedrooms • 1 living ro
10437682	VOVO PREMIER HOTEL	VOVO PREMIER HOTEL.4 out of 5 stars.8.1 Very good 1189 reviews.Gandhi nagar • 2.7 km from centre.Mobile-only price. Hotel room : 1 bed.Original price 444 USD. Curren
9200444	SPLENDID SHUBHAM SERVICED APARTMENTS	SPLENDID SHUBHAM SERVICED APARTMENTS.4 out of 5 for property rating.8.5 Very good 112 reviews.9 km from centre.Mobile-only price. Entire apartment – 121 m^2 : 2 be
11617151	Treebo Global Stay, Jayanagar	Treebo Global Stay, Jayanagar.3 out of 5 stars.7.4 Good 85 reviews.Jayanagar • 6 km from centre.Mobile-only price. Hotel room : 3 beds.Original price 1233 USD. Current pr
8994142	FabHotel Greenwood Xpress Inn - Nr Manipal Hospital	FabHotel Greenwood Xpress Inn - Nr Manipal Hospital.3 out of 5 stars.9.0 Superb 914 reviews.5 km from centre.Early 2025 Deal. Hotel room : 1 bed.Original price 333 USD.
5985890	Tranquil Orchid Serviced Apartments	Tranquil Orchid Serviced Apartments.3 out of 5 for property rating.8.3 Very good 65 reviews.Electronic City • 11 km from centre. Private room : 3 beds • 2 bathrooms.247 U
8512951	Pinnacle Serviced Apartments	Pinnacle Serviced Apartments.3 out of 5 for property rating.8.5 Very good 69 reviews.9 km from centre. Entire apartment – 167 m^2 : 2 beds • 2 bedrooms • 3 living rooms •
10308421	Hotel Sai Bliss International, Near Manyata Tech Park	Hotel Sai Bliss International, Near Manyata Tech Park.8.0 Very good 30 reviews.10 km from centre. Hotel room : 1 bed.Original price 245 USD. Current price 201 USD..Includ
1116332	ibis Bengaluru City Centre - An Accor Brand	ibis Bengaluru City Centre - An Accor Brand.4 out of 5 stars.7.0 Good 1477 reviews.1.3 km from centre. Hotel room : 1 bed.736 USD.Includes taxes and charges.
12947241	Oriva Freshco	Oriva Freshco.3 out of 5 stars.10 Exceptional 3 reviews.11 km from centre.Limited-time Deal. Hotel room : 1 bed.Original price 533 USD. Current price 256 USD..Includes tax
9493436	FabHotel Rotano Grand - Nr Marathahalli Bridge	FabHotel Rotano Grand - Nr Marathahalli Bridge.3 out of 5 stars.8.8 Fabulous 608 reviews.12 km from centre.Early 2025 Deal. Hotel room : 1 bed.Original price 355 USD. Cu
9060886	Zenith Hotels Hebbal Bangalore	Zenith Hotels Hebbal Bangalore.3 out of 5 stars.7.8 Good 142 reviews.10 km from centre.Mobile-only price. Hotel room : 1 bed.Original price 938 USD. Current price 801 US
10594212	Olive Hotel JP Nagar by Embassy Group	Olive Hotel JP Nagar by Embassy Group.3 out of 5 stars.7.8 Good 201 reviews.JP Nagar • 8 km from centre.Mobile-only price. Hotel room : 1 bed.Original price 545 USD. Cur
8985840	Windfall Service Apartment	Windfall Service Apartment.4 out of 5 for property rating.9.0 Superb 27 reviews.Yelahanka • 12 km from centre. Entire apartment – 93 m^2 : 4 beds • 2 bedrooms • 1 living ro



Description	Ranking Position	Rating
Limited-time Deal. Hotel room : 1 bed.Original price 354 USD. Current price 170 USD..Includes taxes and charges.Free cancellation.No prepayment needed.	0	7.9
rb 41 reviews.10 km from centre. Entire apartment – 82 m^2 : 1 bed • 1 bedroom • 1 bathroom.Original price 355 USD. Current price 257 USD..Includes taxes and charges.Free cancellation.No prepayment need	1	9.1
centre. Entire apartment – 36 m^2 : 1 bed • 1 bedroom • 1 living room • 1 bathroom.514 USD.Includes taxes and charges.	2	9.5
room : 1 bed.196 USD.Includes taxes and charges.Free cancellation.No prepayment needed.	3	8.5
. Entire apartment – 102 m^2 : 2 beds • 2 bedrooms • 1 living room • 2 bathrooms.Original price 503 USD. Current price 453 USD..Includes taxes and charges.Free cancellation.No prepayment needed.	4	8.2
-only price. Entire apartment – 111 m^2 : 2 beds • 2 bedrooms • 1 living room • 2 bathrooms.Original price 566 USD. Current price 509 USD..Includes taxes and charges.Free cancellation.No prepayment neede	5	7.8
re. Entire apartment – 102 m^2 : 3 beds • 2 bedrooms • 1 living room • 2 bathrooms.297 USD.Includes taxes and charges.Free cancellation.No prepayment needed.	6	6.9
ile-only price. Hotel room : 1 bed.Original price 444 USD. Current price 399 USD..Includes taxes and charges.Free cancellation.No prepayment needed.	7	8.1
from centre.Mobile-only price. Entire apartment – 121 m^2 : 2 beds • 2 bedrooms • 1 living room • 2 bathrooms.Original price 490 USD. Current price 441 USD..Includes taxes and charges.Free cancellation.	8	8.5
ly price. Hotel room : 3 beds.Original price 1233 USD. Current price 450 USD..Includes taxes and charges.Free cancellation.No prepayment needed.	9	7.4
tre.Early 2025 Deal. Hotel room : 1 bed.Original price 333 USD. Current price 177 USD..Includes taxes and charges.Free cancellation.No prepayment needed.	10	9
• 11 km from centre. Private room : 3 beds • 2 bathrooms.247 USD.Includes taxes and charges.Free cancellation.No prepayment needed.	11	8.3
re apartment – 167 m^2 : 2 beds • 2 bedrooms • 3 living rooms • 2 bathrooms.566 USD.Includes taxes and charges.Free cancellation.No prepayment needed.	12	8.5
om : 1 bed.Original price 245 USD. Current price 201 USD..Includes taxes and charges.Free cancellation.	13	8
l room : 1 bed.736 USD.Includes taxes and charges.	14	7
1 bed.Original price 533 USD. Current price 256 USD..Includes taxes and charges.Free cancellation.No prepayment needed.	15	10
e.Early 2025 Deal. Hotel room : 1 bed.Original price 355 USD. Current price 188 USD..Includes taxes and charges.Free cancellation.No prepayment needed.	16	8.8
Hotel room : 1 bed.Original price 938 USD. Current price 801 USD..Includes taxes and charges.Free cancellation.	17	7.8
Mobile-only price. Hotel room : 1 bed.Original price 545 USD. Current price 425 USD..Includes taxes and charges.Free cancellation.No prepayment needed.	18	7.8
tre. Entire apartment – 93 m^2 : 4 beds • 2 bedrooms • 1 living room • 2 bathrooms.695 USD.Includes taxes and charges.	19	9

APPENDIX-C

ENCLOSURES

The Sustainable Development Goals

SUSTAINABLE DEVELOPMENT GOALS
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS

Sustainable Development Goals (SDGs) Addressed by the Project:

Google Search Using Robotic Process Automation

SDG 8 Decent Work and Economic Growth:

Automation in retrieval of hotel data will help stimulate inclusive and sustainable economic growth, employment, and financial opportunities with efficiency and precision in the hospitality industry, ultimately supporting job development and economic prosperity.

Directly Related SDGs:

SDG 9. Industry, Innovation, and Infrastructure

Based on RPA and APIs, promote agile and resilient systems for effective data retrieval and processing. It contributes to the hospitality ecosystem's robustness and technologically advanced structure.

SDG 10: Reduced Inequalities

Give individuals as well as businesses easy and equal access to precise and current hotel information, which reduces information barriers and enhances inclusiveness in both the travel and hospitality industry.

SDG 17: Partnerships for the Goals

Ensure the facilitation of collaboration between the technology developer, hoteliers, travel agencies, and users toward establishing sustainable and efficient data retrieval practices. It means a coherent, integrated hospitality ecosystem developed in the process.

Secondarily Linked SDGs

SDG 1: No Poverty

Create affordable and effective data retrieval systems for low-income groups and small-scale enterprises, so that such underserved communities and small enterprises are included within the travel and hospitality industry, hence increasing their economic opportunities while decreasing poverty levels.

SDG 4: Quality Education

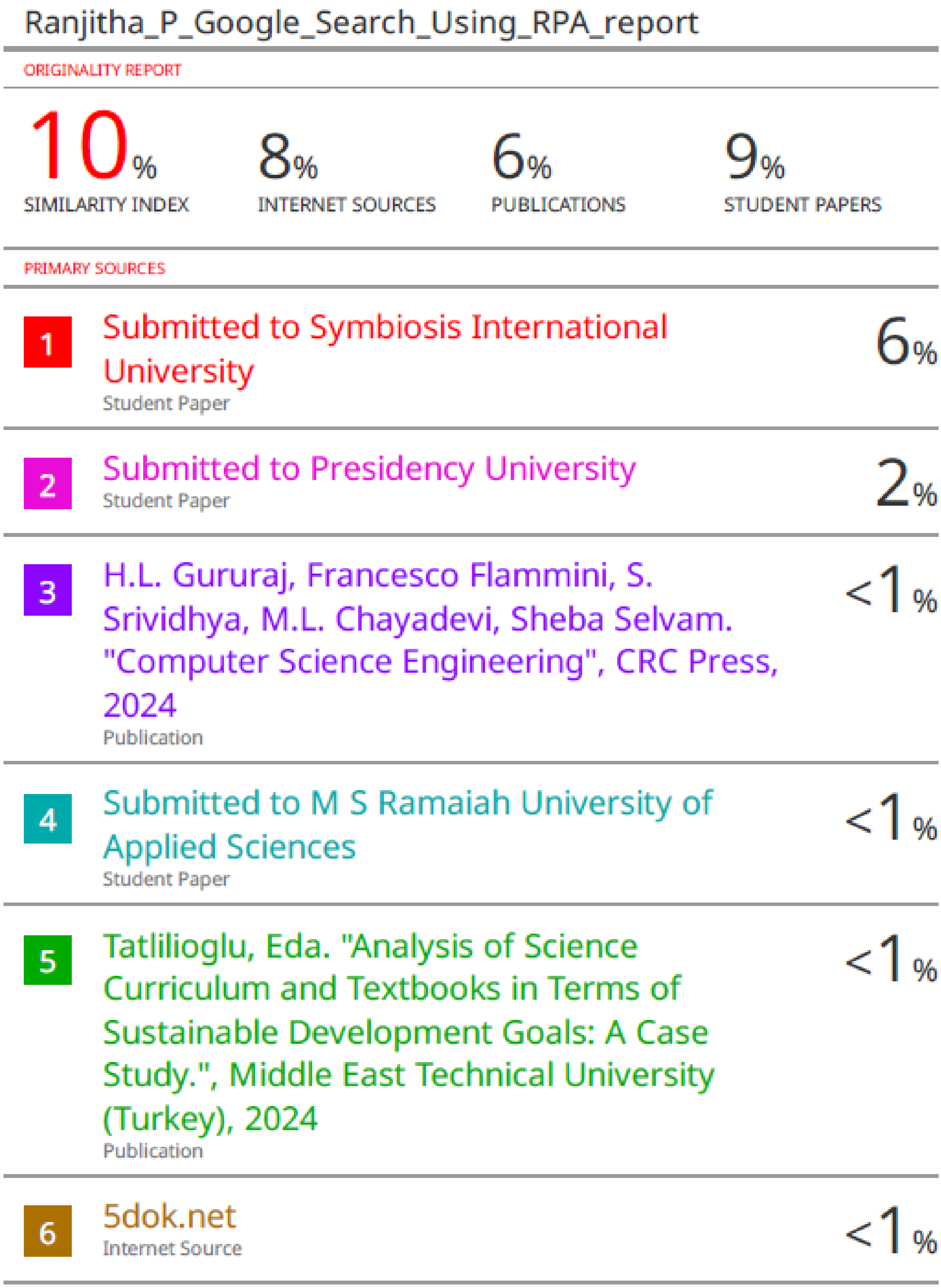
Teach the users on the advantages and use of RPA in data extraction, automation, and hospitality. This will increase the knowledge of users about finances and technology hence quality education.

SDG 12: Responsible Consumption and Production

Improve the efficient usage of resources during data extraction and processing by the automation of the same repetitive task. This enhances sustainability in hospitality due to less human effort and more productivity.

SDG 13: Climate Action

Encourage the adoption of energy-efficient automation systems and sustainable data retrieval practices. This helps reduce the carbon footprint of manually processing data and supports climate action.



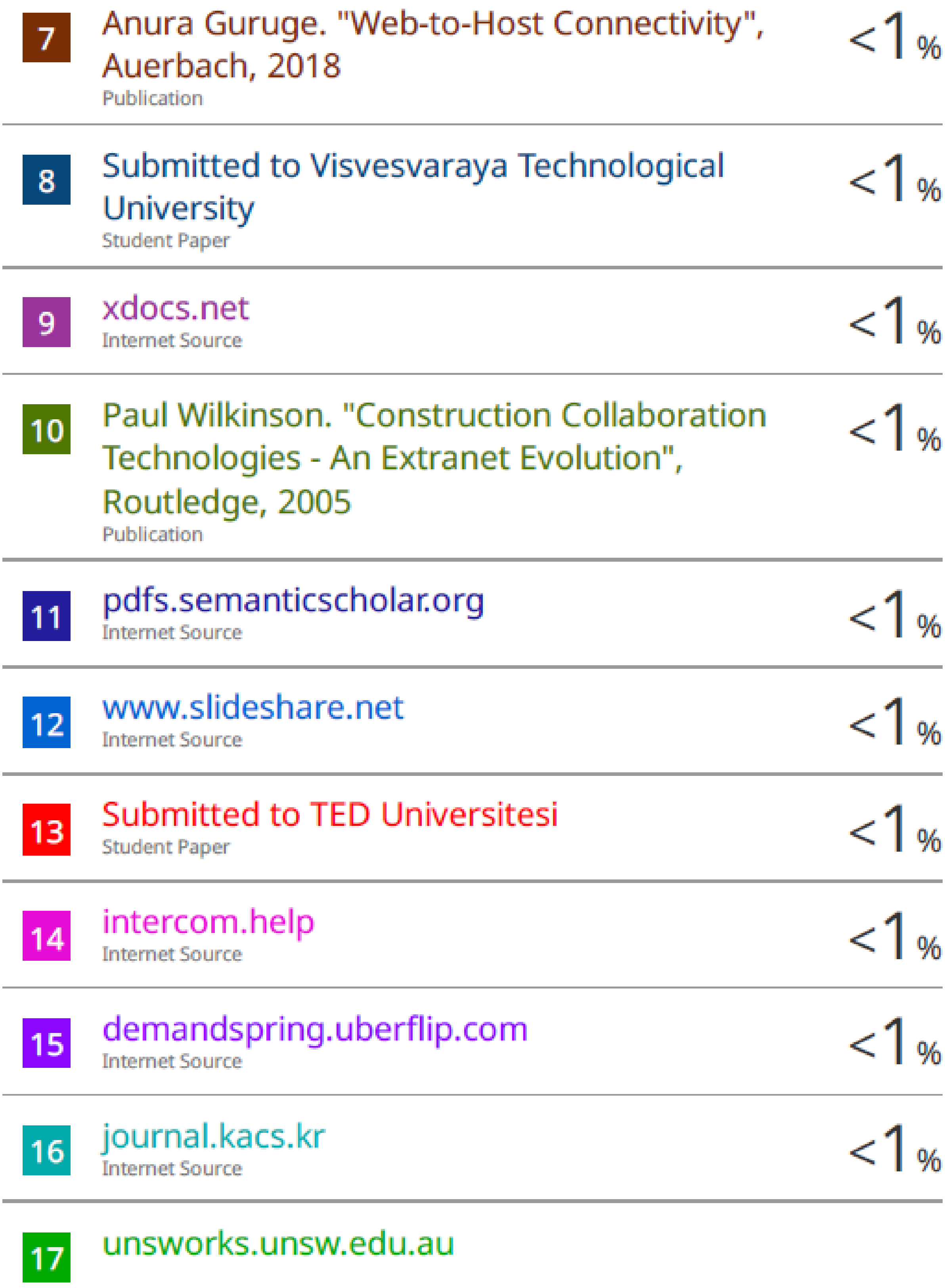
Ranjitha_P_Google_Search_Using_RPA_report

ORIGINALITY REPORT

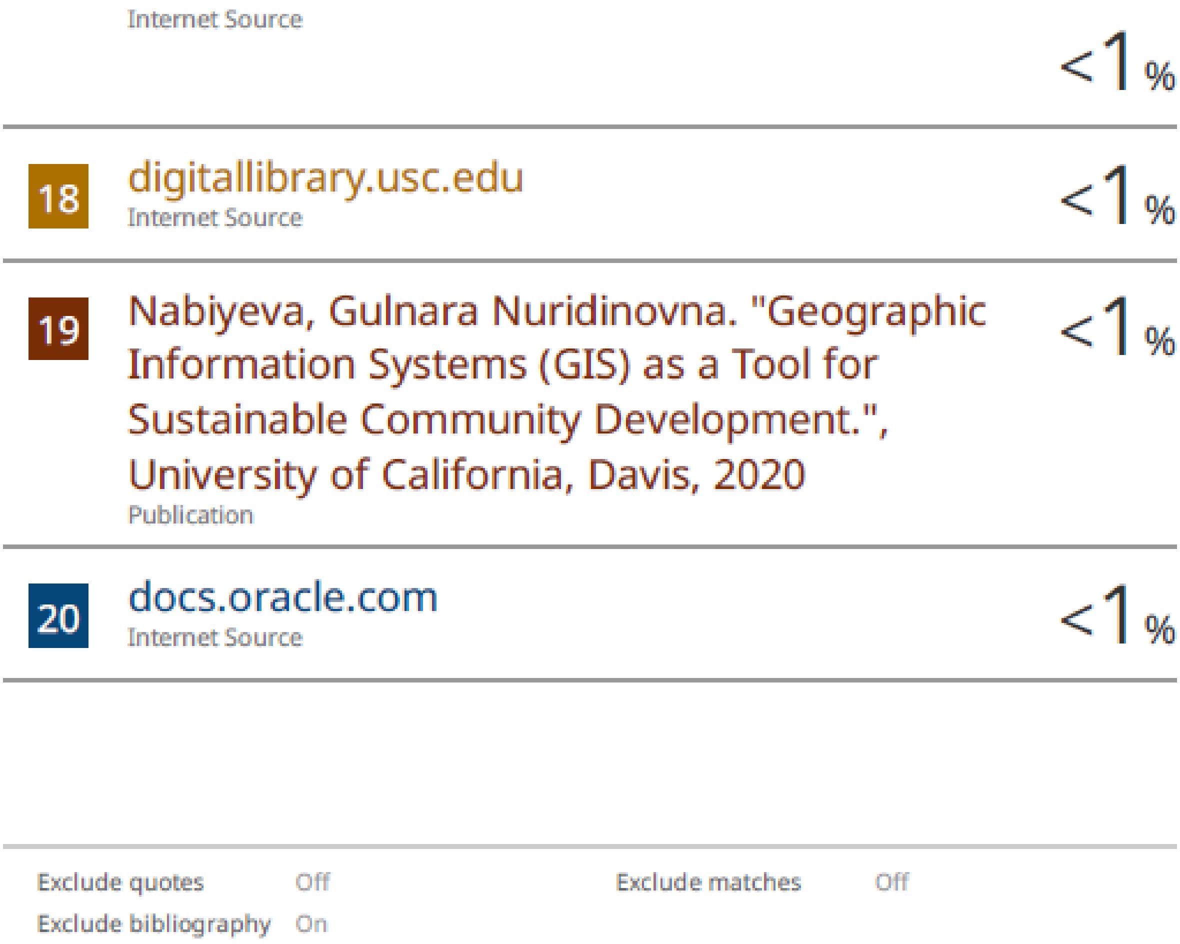
10%	8%	6%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Symbiosis International University Student Paper	6%
2	Submitted to Presidency University Student Paper	2%
3	H.L. Gururaj, Francesco Flammini, S. Srividhya, M.L. Chayadevi, Sheba Selvam. "Computer Science Engineering", CRC Press, 2024 Publication	<1%
4	Submitted to M S Ramaiah University of Applied Sciences Student Paper	<1%
5	Tatlilioglu, Eda. "Analysis of Science Curriculum and Textbooks in Terms of Sustainable Development Goals: A Case Study.", Middle East Technical University (Turkey), 2024 Publication	<1%
6	5dok.net Internet Source	<1%



7	Anura Guruge. "Web-to-Host Connectivity", Auerbach, 2018 Publication	<1 %
8	Submitted to Visvesvaraya Technological University Student Paper	<1 %
9	xdocs.net Internet Source	<1 %
10	Paul Wilkinson. "Construction Collaboration Technologies - An Extranet Evolution", Routledge, 2005 Publication	<1 %
11	pdfs.semanticscholar.org Internet Source	<1 %
12	www.slideshare.net Internet Source	<1 %
13	Submitted to TED Universitesi Student Paper	<1 %
14	intercom.help Internet Source	<1 %
15	demandspring.uberflip.com Internet Source	<1 %
16	journal.kacs.kr Internet Source	<1 %
17	unsworks.unsw.edu.au	



	Internet Source	<1 %
18	digitallibrary.usc.edu Internet Source	<1 %
19	Nabiyeva, Gulnara Nuridinovna. "Geographic Information Systems (GIS) as a Tool for Sustainable Community Development.", University of California, Davis, 2020 Publication	<1 %
20	docs.oracle.com Internet Source	<1 %

Exclude quotes Off

Exclude bibliography On

Exclude matches Off