| **SI. No** | **Contents** | **Page No** |
| --- | --- | --- |
| 1. | COMPANY PROFILE  1.1 OVERVIEW OF THE COMPANY  1.2 VISION AND MISSION OF THE ORGANIZATION  1.3 ORGANIZATION STRUCTURE  1.4 ROLES AND RESPONSIBILITIES OF PERSONNEL IN THE ORGANIZATION  1.5 PRODUCTS AND MARKET PERFORMANCE | 1-6 |
| 2. | ASSESSMENT OF ON JOB TRAINING - 1 2.1 INTRODUCTION  2.2 AIM  2.3 OBJECTIVE  2.4 PURPOSE  2.5 SCOPE  2.6 DISADVANTAGES & ADVANTAGES  2.6.1 ADVANTAGES  2.6.2 DISADVANTAGES  2.7 USE CASE DIAGRAM  2.8 ACTIVITY DIAGRAM  2.9 MODULE  2.9.1 MACHINE LEARNING  2.9.1.1 ALGORITHMS USED  2.9.1.2 LIBRARIES USED: 2.10 REQUIREMENT SPECIFICATION 2.10.1 HARDWARE REQUIREMENTS  2.10.2 SOFTWARE REQUIREMENTS  2.10.3 LANGUAGES USED  2.11CONCLUSION |  |
| 3. | DISCUSSION ON JOB TRAINING CASE-2 | 11-16 |
| 4. | RESUME OF STUDENT | 17-21 |
| 5 | PHOTO GALLERY | 22-25 |

**CHAPTER – 1**

## **COMPANY PROFILE**

**COMPANY PROFILE**

#### **Overview of the company**

Codelab System is a rapidly growing company in the field of computer application Implementation, solutions and services. Codelab System is a service provider of Web-based Development & Web based Software Development Solutions, Mobile Application Development, Graphic Design and Windows Applications. Codelab Systems is headquartered in Mangalore, with the Business development in UAE,Sudi Arabia and Qatar In a short span of 8+ years, our products as well as services & Solutions have been widely accepted by the global market. Today, Codelab Systems has the experience to undertake any IT development or deployment works on a single point responsibility basis. Our efficient and experienced team is greatest resource Intellect’s infrastructure Houses A-team of young and competitive professionals having experience n Web Designing and Software Development who are dedicated to providing high-end Solution to our clients. We develop software and web based applications with Latest Technologies. For web development projects, we also provide hosting And, domain Facility for customers, so they don’t need to bother about that. Our , products and services are user friendly with easy controls and are of Superior specifications. We are always proactive to fulfill client’s needs and requirements to the best possible extent Of their satisfaction. We manage interactive sessions with clients throughout the Project development

### **VISION AND MISSION OF THE ORGANIZATION**

**VISION:**

To help people and businesses throughout the world realize their full potential.Codelab inspiring vision statement seeks to support people. You can see its intention isn’t about business; it’s about people and giving those people the services to be their best selves. With this aim, Codelab has numerous initiatives. It’s a big supporter of inclusivity, diversity, environmental issues, and corporate responsibility. We are on a journey to be the trusted performance leader that unleashes the potential of data.

**MISSION:**

“To enable people and businesses throughout the world to realize their full  
 potential and to organize the world’s information and make it universally

accessible and useful.”

Codelab Systems provides customized package to suit the needs of every client and take into consideration the needs and requirements of each clients and plan different ideas to improve client’s business strategies. Every customer satisfaction is our business and we pay special attention to each clien. To provide best services. The main goal of our company is to provide best and innovative products that will help to drive potential

customers to their businesses

### **ORGANIZATION STRUCTURE**

An organization is a group of people who work together, like a neighborhood association, a charity, a union, or a corporation. You can use the word organization to refer to group or business, or to the act of forming or establishing something. Organizational structure (OS) is the systematic arrangement of human resources in an organization so as to achieve common business objectives. It outlines the roles and responsibilities of every member of the organization so that work and information flow seamlessly, ensuring the smooth functioning

of anorganization.

**Types of Organizational Structure**  
 • Hierarchical  
 • Flat  
 • Flatarchy  
 • Functional  
 • Divisional  
 • Matrix

In a flatarchy, there are little to no levels of management. A company using this structure could have only one manager in between its executive and all other employees. It is called a flatarchy because it is a hybrid of a hierarchy and a flat organization. This type of organizational structure is used more by smaller companies since they have fewer employees, though it can be used in companies of all sizes. While some companies grow out of this organizational structure, others continue to use it. Codelab systems have a Matrix organization structure, where teams report to multiple leaders. The matrix design keeps open

communication between teams and can help companies create more innovative products and services. Using this structure prevents teams from needing to realign every time a new project begins.

### **ROLES AND RESPONSIBILITIES OF PERSONNEL IN THE ORGANIZATION**

We have an expertise team that offer unique solutions. All the members of our team  
are professional, experienced and have in depth knowledge of the technology. Codelab  
Systems provides customized package to suit the needs of every client and take into  
consideration the needs and requirements of each clients and plan different ideas to  
improve client’s business strategies.

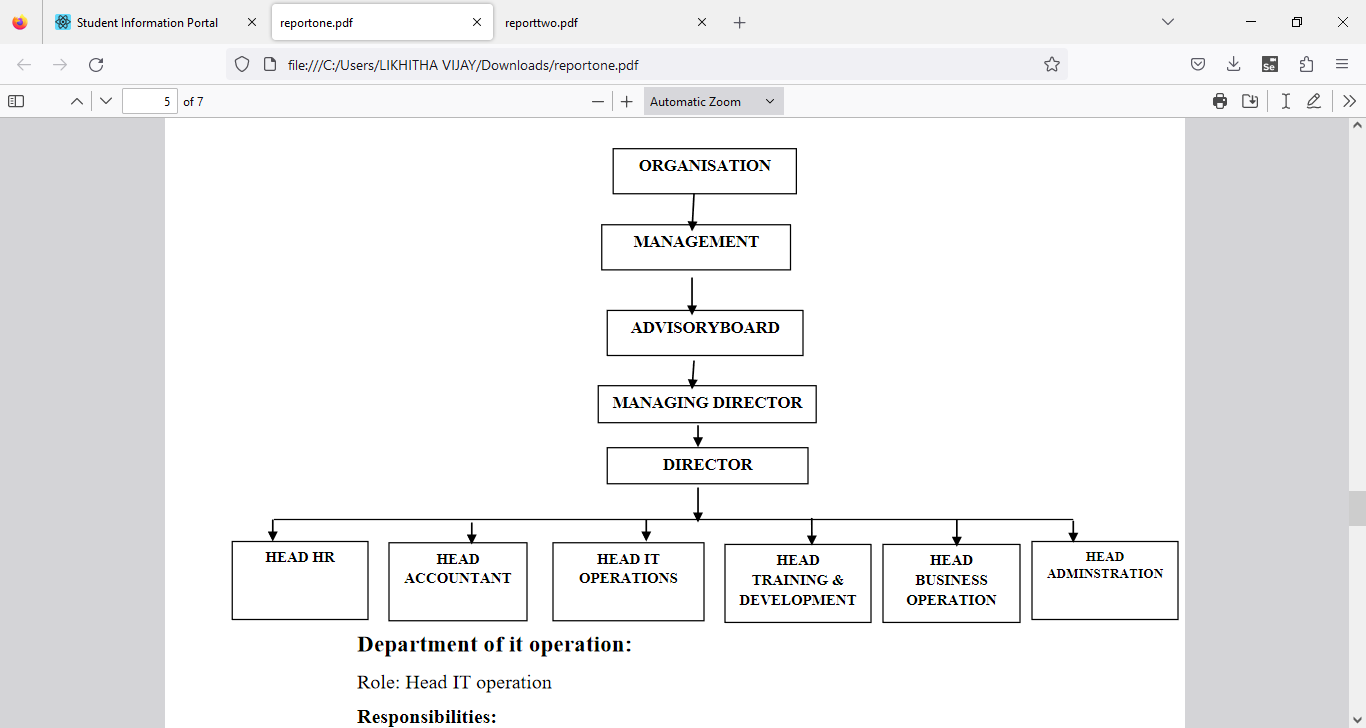


Figure 1.1 Organization Structure

**Department of it operation:  
Role:** Head IT operation  
**Responsibilities:**  
• Development of clients line project  
• Assigning tasks to the subordinate developers  
• Managing client meetings and maintaining a good relationship on stakeholde

**Department of Training and Development:**  
**Role:** Head of Training and Development  
**Responsibilities:**  
• Training to interns and newly joined employees & Work with new projects and

domain

**Department of HR:  
Role:** Head of HR  
**Responsibilities:**

• Maintaining Employees data & payroll calculations  
• Employee leave management & Interns internship program management.

**Department of Account:**  
**Role**: Head of Account.  
**Responsibilities:**  
• Keep track of daily Account & Maintaining Balance sheet and Income tax  
procedure

**Department of administration:**  
**Role:** Head Administration  
**Responsibilities:**  
• All the administration work such as file management, print, maintains data of  
computers and items. Arrangement of training program schedule.

**Department of Business operation:**

**Role:** Head Administration  
**Responsibilities:**  
• Conducting market research, Contact and approach clients for live projects.  
• Communication with new clients and maintaining and managing social

### **PRODUCTS AND MARKET PERFORMANCE**

• MSS LODGE(INDIA) : MSS LODGE is a budget property located in the beautiful  
 city of Ujire.  
• SESCO : SESCO is one of the first enterprises in the electrical equipment sector,  
• QACADEMIA : Q-Academia providing wide range of career oriented IT Courses.

**SERVICES :**We believe in quality services

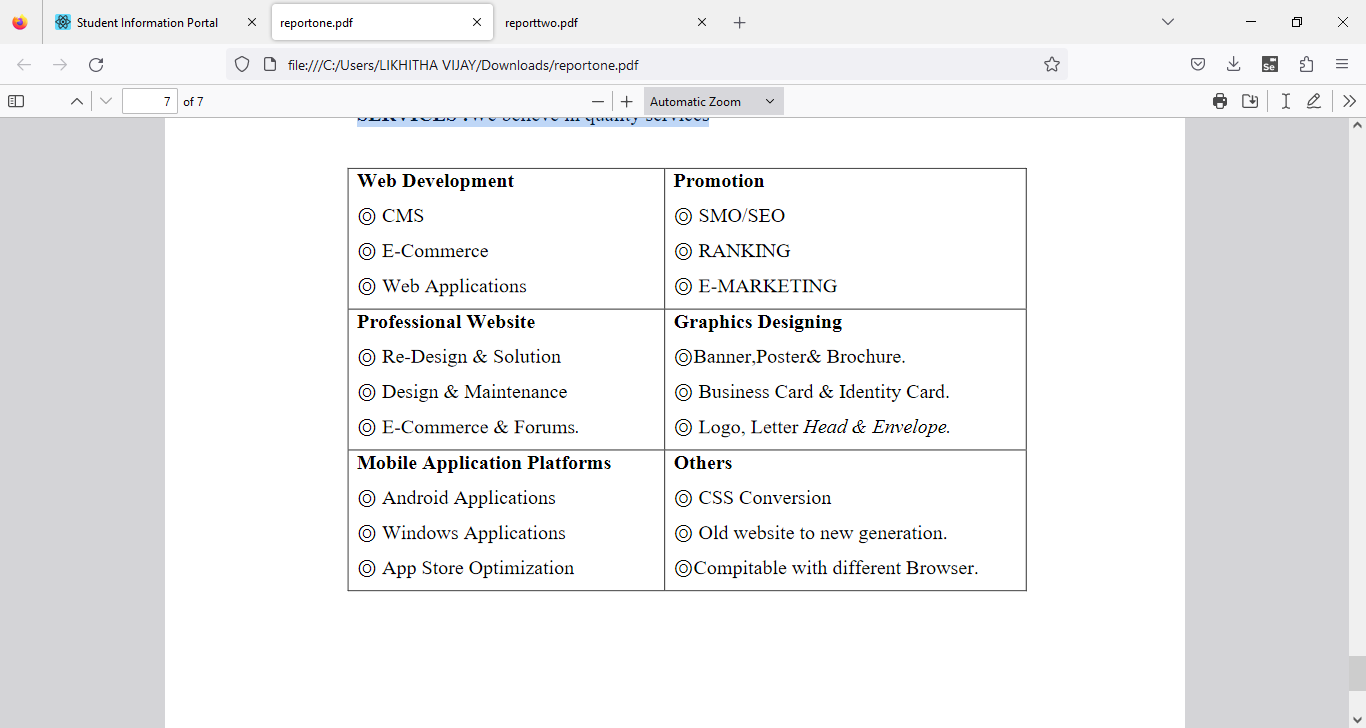


Figure 1.2 Product And Market performance

**CHAPTER- 2**

## **ASSESSMENT OF ON JOB TRAINING - 1**

## **ASSESSMENT OF ON JOB TRAINING - 1**

## **CASE-1 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

* 1. **INTRODUCTION:**

In recent years, urban tourism becomes more and more popular. With the help of E-commerce, it is convenient to choose a sight-seeing destination through Internet. But here comes a typical question for a trip: if we plan to visit the Forbidden City in Beijing for several days, where should we live? Selecting a suitable hotel can be vital for a pleasant trip. In general, this question corresponds to recommending a hotel given a certain destination. However, the past hotel check-in data of an individual may be sparse due to low rating frequency, which means the performance of rating based collaboration filtering technique is poor. Moreover, it’s important to serve for newly registered customers who has no hotel booking history [1]. The sparsity issue and the so-called cold start problem are the main challenges for hotel recommendation. This paper proposes a novel framework to solve these issues. First, CF method is combined with CBF method to overcome sparsity issue, while guaranteeing high enough accuracy. Existing recommender systems basically adopt two kinds of methods: collaborative filtering (CF) or contentbased filtering (CBF). CF focuses on finding similar users based on user-item rating matrix, the precision of which has proven to be good but sensitive to sparse data. On the other hand, CBF manages to match the content of item with users’ interest, the precision of which is usually not well since it depends on the quality of extracted features. To overcome the drawback of these two methods, an idea of combining them together is then proposed. Next, we notice that the intent of trip serves as a vital factor when selecting a hotel. For example, those users on a business trip are more concerned about noise while those on a family trip care more about service and facilities. In general, people who have the same intent share similar preference. Such information is easy to acquire and effective for cold start users. For this reason, the travel intent are introduced into the proposed framework for solving cold start problem. Furthermore, one’s preference for hotels tends to be complicated. When recommender system encounters a new user, background knowledge may be insufficient. It is hard to decide whether specific hotel match his/her preference o not. One possible solution is to employ diversity techniques [2], to satisfy one’s preference as broad as possible, so that monotony of the recommendation result is avoided. Our contributions are listed as follows: • We focus on user preference analysis and solve sparsity issue by integrating CF and CBF. • The intent of a trip is introduced to solve cold start problem with higher prediction accuracy. • We use diversity techniques to optimize the hotel recommendation list. • Experiments show that the proposed hotel recommendation framework outperforms classical approaches. The rest of the paper is organized below. In Section II, we summarize the existing techniques for recommendation system and possible solutions to handle sparsity issue. Next, the hotel recommendation framework is proposed in Section III. Then, experiments are conducted to evaluate the performance of our framework.

### **AIM:**

Based on the information provided, it seems like the aim of this concept is to address the challenges of hotel recommendation in the context of urban tourism, particularly focusing on the issues of sparsity in hotel check-in data and the cold start problem for newly registered customers. The proposed framework aims to combine collaborative filtering (CF) and content-based filtering (CBF) methods to overcome sparsity, integrate the travel intent to address the cold start problem, and utilize diversity techniques to optimize hotel recommendations.

### **1.3 OBJECTIVE:**

The primary objectives of this concept can be summarized as follows:

* Addressing Sparsity Issue: Integrating CF and CBF methods to improve recommendation
* accuracy despite sparse data.

Handling Cold Start Problem: Introducing travel intent as a factor in recommendation to improve accuracy for newly registered users without hotel booking history.

* Optimizing Recommendation Diversity: Employing diversity techniques to ensure recommendation lists cater to a broad range of user preferences, avoiding monotony.
* Evaluation and Comparison: Conducting experiments to evaluate the performance of the proposed framework against classical approaches, demonstrating its superiority.

### **PURPOSE:**

The purpose of this concept is to propose a novel framework for hotel recommendation in the context of urban tourism, addressing the challenges of sparsity in hotel check-in data and the cold start problem for newly registered customers. The main objectives include:

* Integration of CF and CBF Methods: Combining collaborative filtering (CF) and content-based filtering (CBF) methods to overcome the sparsity issue in hotel recommendation systems, ensuring high accuracy even with limited historical data.
* Incorporation of Travel Intent: Recognizing the importance of travel intent in hotel selection and leveraging it as a vital factor in recommendation systems. By understanding the purpose of the trip (e.g., business or family), the framework aims to provide more accurate recommendations tailored to users' preferences.
* Utilization of Diversity Techniques: Employing diversity techniques to enhance the recommendation list's breadth and avoid monotony, especially when dealing with new users with limited background knowledge. By offering a diverse set of options, the framework aims to cater to a wide range of user preferences.
* Evaluation and Validation: Conducting experiments to assess the performance of the proposed framework against traditional approaches. By demonstrating its superiority in terms of recommendation accuracy and user satisfaction, the framework aims to establish its effectiveness in real-world scenarios.

### **SCOPE:**

The framework proposes the integration of collaborative filtering (CF) and content-based filtering (CBF) methods to overcome the sparsity issue in hotel recommendation systems. By leveraging both user-item interactions and item features, the framework aims to enhance recommendation accuracy Recognizing the significance of travel intent in hotel selection, the framework introduces the concept of trip intent as a crucial factor in recommendation systems. By understanding the purpose of the trip (e.g., business or family), the framework seeks to provide more personalized and relevant recommendations tailored to users' preferences. The framework employs diversity techniques to optimize the hotel recommendation list and avoid monotony in recommendations. By offering a diverse set of options, the framework aims to cater to a wide range of user preferences and enhance the overall user experience.

Evaluation and Validation: The proposed framework undergoes rigorous experimentation to evaluate its performance and compare it against classical approaches. Through empirical validation, the framework aims to demonstrate its superiority in terms of recommendation accuracy and user satisfaction.

### 

### 

### **DISADVANTAGES & ADVANTAGES: 1.6.1 ADVANTAGES:**

The proposed framework for hotel recommendation presents several advantages over traditional approaches:

* Improved Recommendation Accuracy: By integrating collaborative filtering (CF) and content-based filtering (CBF) methods, the framework addresses the sparsity issue inherent in hotel recommendation systems. This integration ensures higher recommendation accuracy, even when dealing with limited past hotel check-in data.
* Enhanced Personalization with Travel Intent: Incorporating travel intent as a key factor in the recommendation process allows for more personalized recommendations. By understanding the purpose of the trip (e.g., business or family), the framework can tailor recommendations to match users' specific preferences and requirements.
* Effective Handling of Cold Start Problem: The inclusion of travel intent helps mitigate the cold start problem for newly registered customers with no hotel booking history. By leveraging the intent of the trip, the framework can make accurate recommendations even for users with limited background information.
* Diverse Recommendation Options: The framework employs diversity techniques to optimize the hotel recommendation list, ensuring a broad range of options that cater to various user preferences. This helps prevent monotony in recommendations and increases user satisfaction by offering a diverse selection of hotels.
* Empirical Validation: Through rigorous experimentation, the framework demonstrates its superiority over classical approaches in terms of recommendation performance. The empirical evidence provided in the paper confirms the effectiveness of the proposed framework in real-world scenarios

### **1.6.2 DISADVANTAGES:**

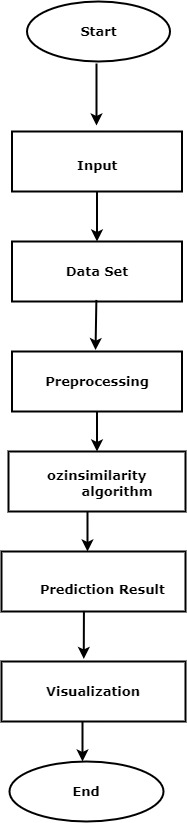
* Complexity of Implementation: Integrating collaborative filtering (CF) and content-based filtering (CBF) methods, along with incorporating travel intent and diversity techniques, can make the framework complex to implement and maintain. This complexity may pose challenges in terms of system development, deployment, and ongoing management.
* Increased Computational Resources: Combining multiple recommendation techniques may require significant computational resources, especially when dealing with large datasets or real-time recommendation scenarios. This could result in increased processing time and resource utilization, impacting system scalability and performance.
* Dependency on Data Quality: The effectiveness of the framework relies heavily on the quality and availability of data, including user preferences, hotel features, and travel intent information. Inaccurate or incomplete data could lead to suboptimal recommendations and undermine the overall performance of the system.
* Difficulty in Parameter Tuning: Optimizing the performance of the framework may require fine-tuning various parameters, such as weighting factors for CF and CBF methods, diversity thresholds, and intent classification algorithms. Finding the optimal parameter settings could be challenging and time-consuming.
* Potential for Overfitting: Incorporating multiple recommendation techniques and features increases the risk of overfitting the model to the training data. This could lead to overly specific recommendations that do not generalize well to new users or contexts.
* User Privacy Concerns: Gathering and analyzing user data, including travel intent and preferences, raises privacy concerns. Users may be reluctant to share personal information, leading to limited availability of data for recommendation purposes and potentially affecting recommendation accuracy.

### **1.7 USE CASE DIAGRAM:**

### 

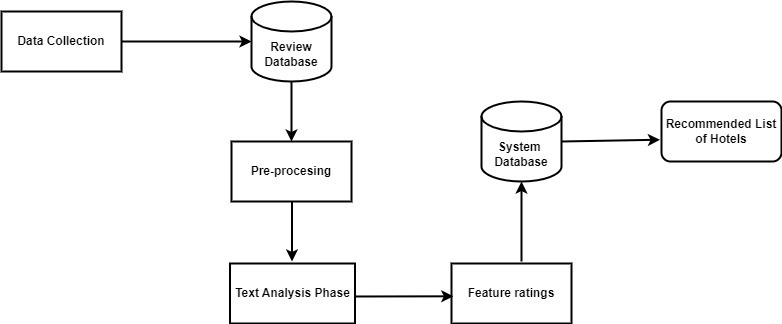
### Fig 1: use case diagram

### **1.8 ACTIVITY DIAGRAM**



### Fig 2: Activity diagram

### **1.9 MODULE:**



### Fig 3: Model diagram

### 

### 1.9.1 **MACHINE LEARNING**

**Machine learning** is a branch of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that enable computers to learn and make predictions or decisions without being explicitly programmed. It involves the use of data to train models, allowing them to identify patterns, extract insights, and make predictions or decisions based on new data. Machine learning algorithms are categorized into supervised, unsupervised, and reinforcement learning, each serving different purposes in tasks such as classification, regression, clustering, and reinforcement learning.

**1.9.1.1 ALGORITHMS USED**

* **OZINSIMILARITY ALGORITHM** :TheOzinSimilarity algorithm is a method for computing similarity between molecules in chemistry. It is named after Professor Geoffrey Ozin, who developed it. The algorithm is primarily used in the field of cheminformatics and molecular informatics to assess the structural similarity between chemical compounds.TheOzinSimilarity algorithm is based on the principle of comparing the structural features of molecules to determine their similarity. It considers various molecular descriptors, such as atom types, bond types, connectivity patterns, and molecular fingerprints, to quantify the similarity between pairs of molecules.

The algorithm typically involves several steps:

* Molecular Representation: Convert each molecule into a standardized representation, often in the form of molecular graphs or fingerprints.
* Descriptor Calculation: Compute molecular descriptors or fingerprints that capture important structural features of the molecules.
* Similarity Calculation: Compare the descriptors or fingerprints of pairs of molecules using a similarity metric, such as Tanimoto coefficient, Dice coefficient, or cosine similarity.
* Thresholding: Optionally, apply a threshold to determine whether two molecules are sufficiently similar based on the computed similarity score

**1.9.1.1 LIBRARIES USED:**

* **PANDAS**: is a Python library widely used in machine learning for data manipulation, preprocessing, and analysis. It provides easy-to-use data structures and functions for loading, exploring, cleaning, and preprocessing structured data. With Pandas, ML practitioners can efficiently handle tasks such as data loading, exploration, feature engineering, preprocessing, integration with ML libraries, and model evaluation. Its versatility and seamless integration with other Python libraries make it an essential tool in the machine learning workflow.
* **NUMPY**: is a fundamental library in Python for numerical computing, widely used in machine learning. It provides efficient data structures and functions for array manipulation, mathematical operations, and integration with other libraries. NumPy multidimensional arrays, mathematical functions, and performance optimizations make it essential for handling large datasets, implementing machine learning algorithms, and performing complexcomputations efficiently.
* **NLTK**: Stands for Natural Language Toolkit. It's a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning. NLTK is intended to support research and teaching in computational linguistics, artificial intelligence, and cognitive science, and to help bridge the gap between these areas and mainstream computer science. It's widely used for tasks like text classification, sentiment analysis, machine translation, and more.
* **STOPWORDS**: Are common words in a language that are typically filtered out or ignored during natural language processing (NLP) tasks because they are considered to have little semantic significance. These words include articles (e.g., "a", "an", "the"), prepositions (e.g., "in", "on", "at"), conjunctions (e.g., "and", "or", "but"), and other frequently occurring words that do not carry much meaning on their own.In NLP, stopwords are often removed from text data before performing tasks such as text analysis, classification, or information retrieval. By filtering out stopwords, the focus can be placed on the more meaningful words in the text, which can improve the accuracy and efficiency of NLP algorithms. StopwordS lists may vary depending on the specific application or domain. Commonly used stopwords are provided in libraries such as NLTK (Natural Language Toolkit) for various languages. However, it's important to note that the choice of stopwords can influence the outcome of NLP tasks, and custom stopwordS lists may be used based on the specific requirements of the task or domain.
* **WordNetLemmatizer:** Is a tool provided by NLTK (Natural Language Toolkit), a popular Python library for natural language processing (NLP). The WordNetLemmatizer is specifically used for lemmatization, which is the process of reducing words to their base or canonical form, known as the lemma. In NLP, lemmatization is often performed to normalize words, reducing them to their dictionary form to improve text analysis and understanding. For example, the lemma of "running" is "run", the lemma of "better" is "good", and the lemma of "mice" is "mouse".
* **FLASK**: is a lightweight web framework for Python, used to build web applications quickly and efficiently. It provides tools and libraries for handling HTTP requests, routing, templating, and more, making it easy to create web applications with minimal boilerplate code. Flask follows a minimalist philosophy, allowing developers to add only the features they need, making it ideal for building small to medium-sized web applications and APIs. Its simplicity, flexibility, and extensive documentation make it a popular choice among developers for prototyping, building MVPs (Minimum Viable Products), and developing RESTful APIs.
* **Word\_tokenize:** is a function commonly used in natural language processing (NLP) libraries such as NLTK (Natural Language Toolkit) or spaCy. It essentially breaks down a piece of text into individual words or tokens, where each token represents a word in the text. This process is a fundamental step in many NLP tasks, such as text classification, named entity recognition, and sentiment analysis. The word\_tokenize function takes a string of text as input and returns a list of tokens, where each token is typically a word or punctuation mark.
* **Literal\_eval** : is a function within the ast (Abstract Syntax Trees) module in Python's standard library. It's used to safely evaluate strings containing Python expressions or literals. When you have a string that represents a Python data structure like a list, dictionary, tuple, or even simple literals like integers or strings, literal\_eval can parse and evaluate it into the corresponding Python object.The main advantage of literal\_eval over using eval is that it's safer. eval can execute arbitrary code, which poses a security risk if the string being evaluated comes from an untrusted source. literal\_eval, on the other hand, only evaluates strings that represent literals or expressions with a simple structure, making it safer to use in situations where security is a concern.

### **1.10 REQUIREMENT SPECIFICATION:**

**1.10.1 HARDWARE REQUIREMENTS**

* RAM: 8 GB or higher
* Storage: 256 GB SSD or higher
* Network: Ethernet/Wi-Fi for internet connectivity
* Display: 15-inch monitor or larger
* Processor: Intel Core i5 or equivalent

**1.10.2 SOFTWARE REQUIREMENTS**

* Operating System: Windows 10 or Ubuntu 20.04 LTS
* Web Browser: Google Chrome or Mozilla Firefox
* Integrated Development Environment (IDE): Visual Studio Code, Python 3.8 or higher installed

**1.10.3 LANGUAGES USED**

* Front-end: HTML, CSS, JavaScript
* Back-end: Python, Flask

### **1.11 CONCLUSION:**

In conclusion, the proposed framework for hotel recommendation addresses key challenges in urban tourism, specifically focusing on the issues of data sparsity and the cold start problem. By integrating collaborative filtering (CF) and content-based filtering (CBF) methods, incorporating travel intent, and employing diversity techniques, the framework offers a comprehensive solution to enhance recommendation accuracy and user satisfaction.

# 

# **CHAPTER -3**

**DISCUSSION ON JOB TRAINING CASE-2**

## **CASE-2:IMPLEMENTATION**

1. **IMPORTING LIBRARIES**

import nltk

nltk.download('wordnet')

import numpy as np

import pandas as pd

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

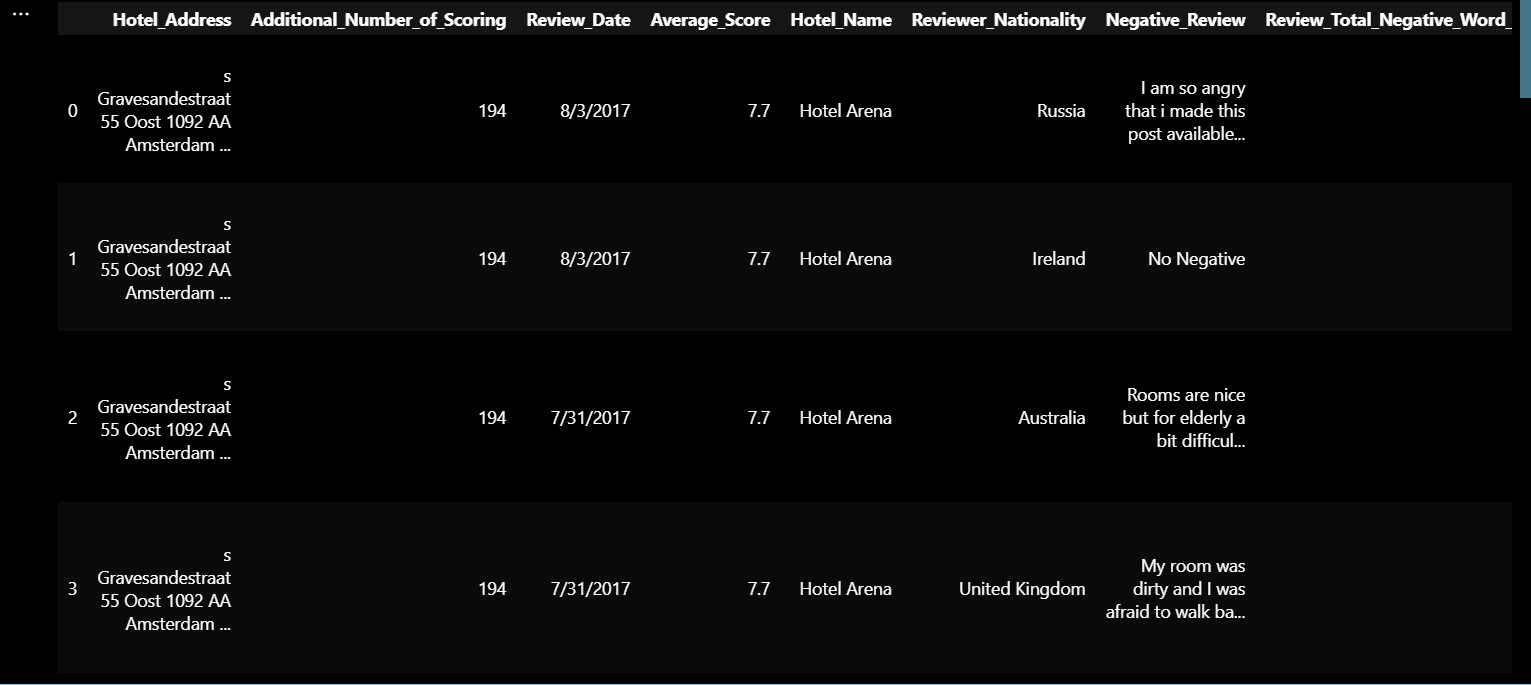
from nltk.stem.wordnet import WordNetLemmatizer

from ast import literal\_eval

data = pd.read\_csv("hotel.csv")

data.head()

**OUTPUT:**



# Replacing "United Kingdom with "UK"

data.Hotel\_Address = data.Hotel\_Address.str.replace("United Kingdom", "UK")

# Now I will split the address and pick the last word in the address to identify the country

data["countries"] = data.Hotel\_Address.apply(lambda x: x.split(' ')[-1])

print(data.countries.unique())

**OUTPUT:**



def impute(column):

column = column[0]

if (type(column) != list):

return "".join(literal\_eval(column))

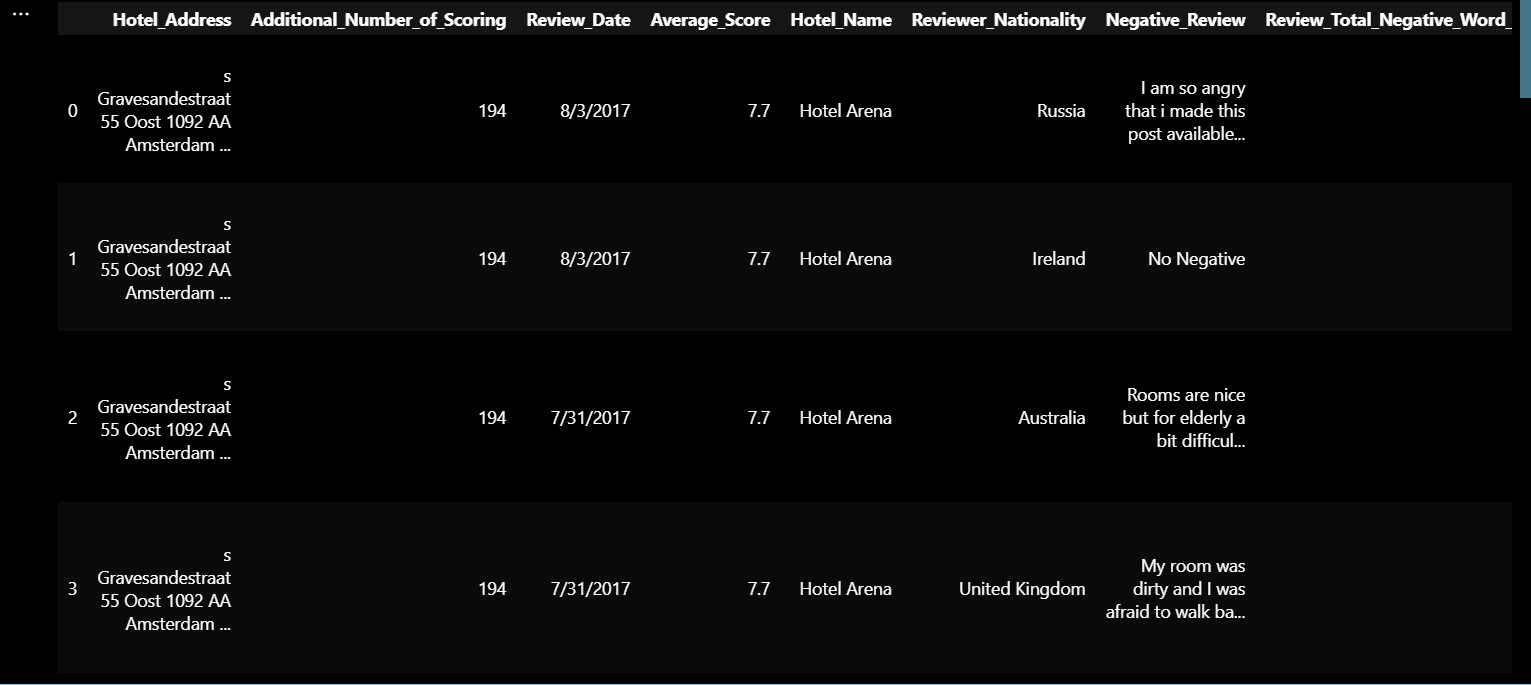
else:

return column

data["Tags"] = data[["Tags"]].apply(impute, axis=1)

data.head()

**OUTPUT:**



data['countries'] = data['countries'].str.lower()

data['Tags'] = data['Tags'].str.lower()

def recommend\_hotel(location, description):

description = description.lower()

word\_tokenize(description)

stop\_words = stopwords.words('english')

lemm = WordNetLemmatizer()

filtered = {word for word in description if not word in stop\_words}

filtered\_set = set()

for fs in filtered:

filtered\_set.add(lemm.lemmatize(fs))

country = data[data['countries']==location.lower()]

country = country.set\_index(np.arange(country.shape[0]))

list1 = []; list2 = []; cos = [];

for i in range(country.shape[0]):

temp\_token = word\_tokenize(country["Tags"][i])

temp\_set = [word for word in temp\_token if not word in stop\_words]

temp2\_set = set()

for s in temp\_set:

temp2\_set.add(lemm.lemmatize(s))

vector = temp2\_set.intersection(filtered\_set)

cos.append(len(vector))

country['similarity']=cos

country = country.sort\_values(by='similarity', ascending=False)

country.drop\_duplicates(subset='Hotel\_Name', keep='first', inplace=True)

country.sort\_values('Average\_Score', ascending=False, inplace=True)

country.reset\_index(inplace=True)

return country[["Hotel\_Name", "Average\_Score", "Hotel\_Address"]].head()

OUTPUT:

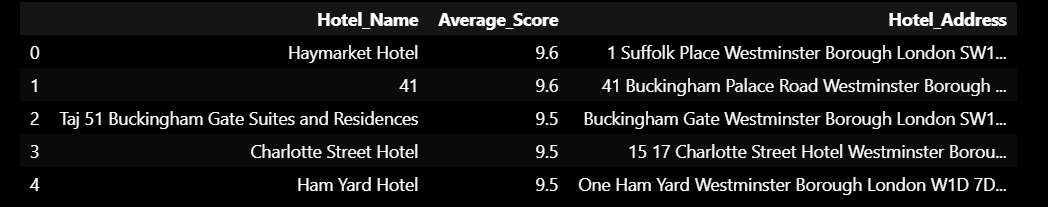
recommend\_hotel('Italy', 'I am going for a business trip')

**OUTPUT:**



recommend\_hotel('UK','I am going on a honeymoon, I need a honeymoon suite room for 3 nights')

**OUTPUT:**



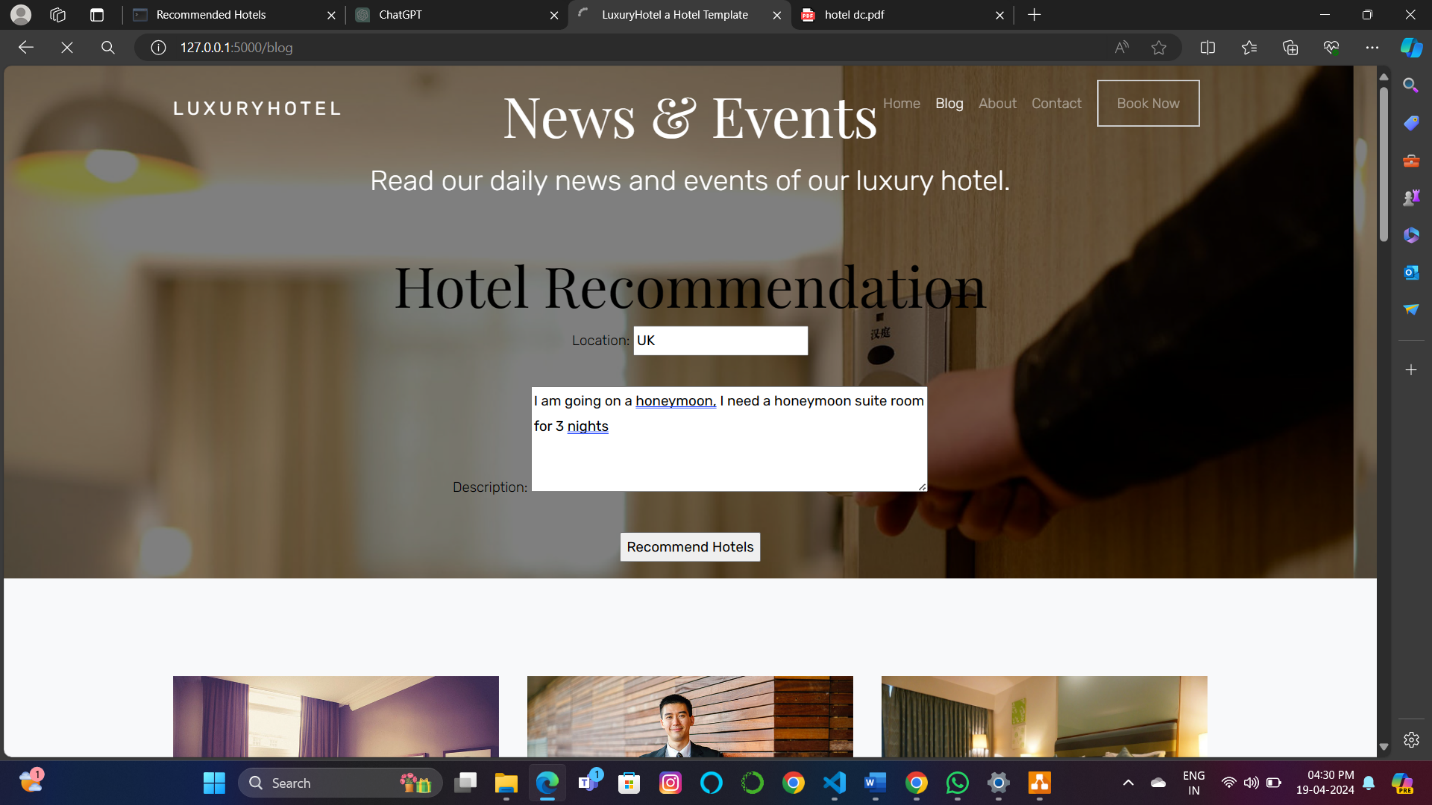
**3.2 UI DESIGNING**

**SCREENSHOT 1: Home Page**



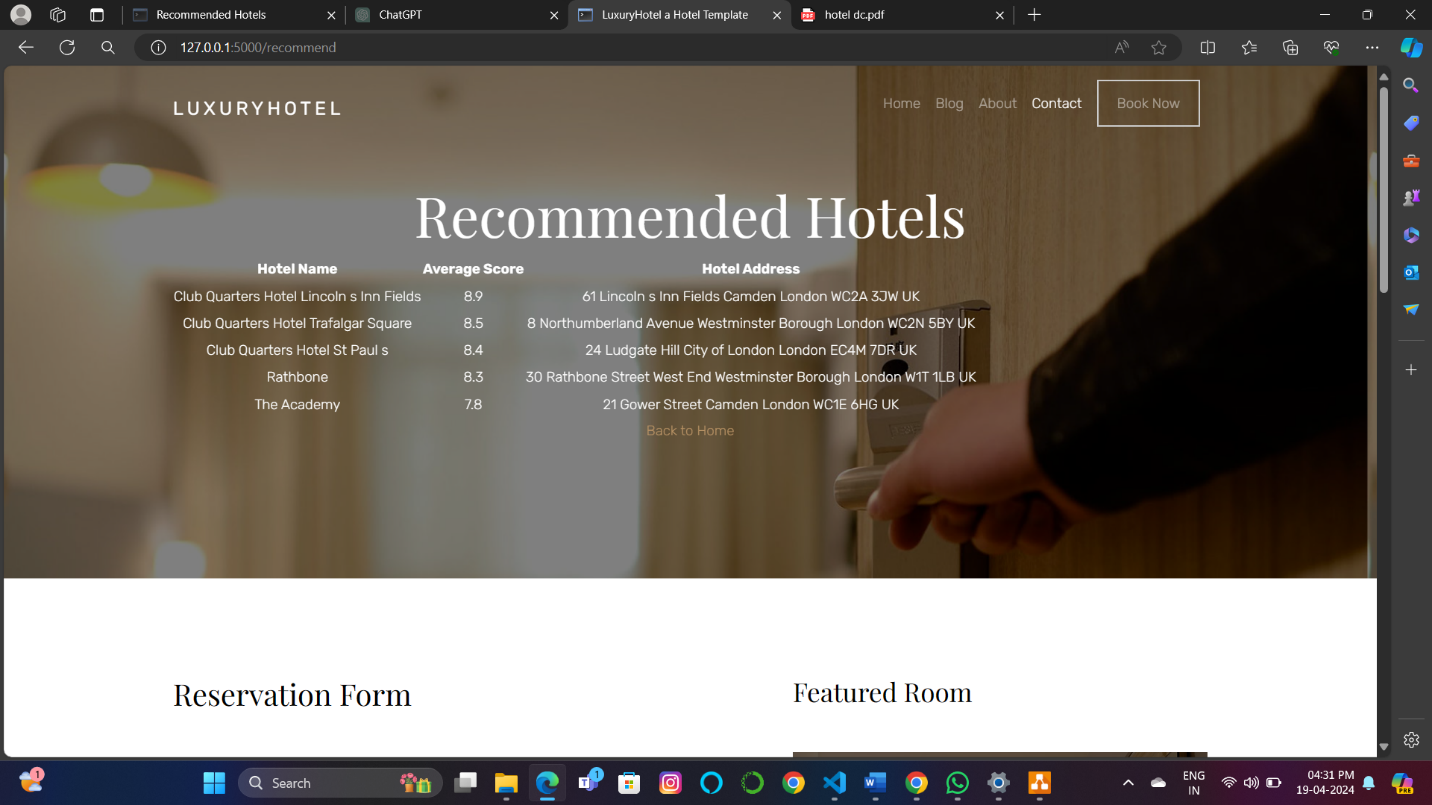
**Figure 1: Home Page**

**SCREENSHOT 2: Input Page**



**Figure 2: Blog Page**

**SCREENSHOT 3: Output Page**



**Figure 3: Book Now Page**

**4. RESUME OF STUDENT**



**5. PHOTO GALLERY**