

Travel Management System: Vacation Nation

A PROJECT REPORT

Submitted by

Vaibhav Singh



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April, 2022

Table of Contents

	Page no.
1. Introduction	
1.1. Existing System.....	3
1.2. Proposed System.....	4
2. Literature Survey	
2.1. Literature Summary.....	4
2.2. Problem Definition	8
3. Overview of the Work	
3.1.Objectives of the Project.....	8
3.2.Software Requirements.....	10
3.3.Hardware Requirements.....	12
4. System Design	
4.1.Algorithms.....	12
4.2. System Modeling.....	15
5. Implementation	
5.1.Description of Modules/Programs.....	21
5.2.Source Code.....	21
5.3.Test cases.....	36
6. Output and Performance Analysis	
6.1.Execution Snapshots.....	37
6.2.Output – in terms of performance metrics.....	40
6.3. Performance comparison with existing works	42
7. Conclusion and Future Directions.....	42
8. References.....	43

ABSTRACT

Nowadays, it has become a normalcy to travel and nearly everyone goes on a vacation or tour. Moreover, with the increase in the standard of living, this trend is only going to be increasing as the years go on. However, with this increase in the tourism industry there also comes various problems which are faced by the travellers and therefore, the Tourism Management System would play a vital role in planning the perfect trip. The tourism management system allows the user access the system at all times and from anywhere, and provides the user with information and reviews about the places they can visit as well as the hotels and restaurants that they can book. Along with this, the system gives the services of an AI chatbot for better user experience and a machine learning based recommendation system for helping the user in creating their travel plan according to their preferences.

1. Introduction

Travel and tourism management system is used to book a tour from anywhere in the world through a single dynamic website that helps the user to know everything about the places and tour details on a single website. The administrator can add packages from certain travel agencies and hotels to the website by creating a tour page. The user can see the confirmation on their booking page. It is the easiest platform for all travellers to easily book and know all the details. Tour Management system is a dynamic website for tourism business. It is a dynamic and responsive web design. It is also called a travel technology solution for agencies and tour operators. Almost everyone goes on vacation because this “Vacation Nation” would play a vital role in planning the perfect trip. The tourism management system allows the system user to access all the details such as location, event etc. The main purpose is to help tourism companies manage customers and hotels etc. The system can also be used for professional and business travel.

1.1 Existing System

In the existing system, every task is done manually and processing is also a tedious work. In the previous system, passengers-maintained schedule details manually with pencil and paper, which was time-consuming and expensive. Travellers are not able to fulfil their need in time

and also the results may not be accurate. Due to manual maintenance, there are many problems and disadvantages in the system. Some of them are:

Disadvantages of the existing system:

- An increased number of transactions leads to an increase in the number of source documents and therefore difficult to maintain.
- If any administrator, user enters wrongly, maintenance will be very difficult.

1.2 Proposed System

The proposed system is designed to be more efficient than the manual system. It invokes all the basic tasks that are now done manually, such as form transactions and reports, which is another advantage. The proposed system is a fully computerized application. Hundreds of records can be searched and displayed without taking any significant time.

Advantages of the proposed system:

- Provides accurate information
- Simplifies manual work
- Minimizes documentation related work
- Provides up-to-date information
- Friendly environment by providing warning messages.
- Reservation confirmation notification

2. Literature Survey

2.1 Literature Summary

SNo.	Title	Authors	Description	Methodology	Limitations
1.	Developing A Creative Travel Management System Based on Software Reuse and Abstraction Techniques	Meiyu Shi, Sicong Ma and Hongji Yang	This paper demonstrates a creative travel plan generation system for creating innovative and special travel plans. In this system, there are three kernel phases to generate	The system first uses Density-Based Clustering algorithms to derive different travel components with a high level of abstraction from the raw data from the characteristic	The system has compatibility issues with low specs devices as well as a need for more research on both the software reuse and creativity metrics system.

			creative travel plans including “Information Abstraction”, “Information Reuse” and “Information Formulation”	query. Then, according to collected and abstracted travelling data, the system generates raw plans-based rules and algorithms of software reuse. Finally, the system builds a creativity metrics system to rank the derived plans based on designed rules.	
2.	Tours and Travel Management System	Kuchekar Rutvik Baban, Mehra Prasad Sanjay and Jadhav Anushka Manoj	This paper describes about the automation of the travel and tourism industry which will allow the users to book a tour or travel plan from anywhere in the world, at anytime, just with the use of a single dynamic website.	The system is completely menu-pushed and is a effective tool for interactive layout with all entry displays being interactive. It automates the booking and creation of the travel plan for the users by making the manual processes online. It uses a standardized cosine match calculation for finding similarities with the time result and may reflect similarities of current users' interest. It also, prevents the user from making an invalid selection and to recognise the range of options to be had.	The system database needs to be updated manually periodically for maintaining accuracy of the algorithms with no automatic updating procedure available.

3.	Intelligent Tourism Management System	Ernest E. Onuiria, Henry C. Omorojeb, Chukwudi G. Ntimac and Ayokunle A. Omotunded	The aim of this research was to create a web-based application designed to allow one to easily access relevant information, make recommendation for places, and make bookings online for travel and accommodation.	It assists tourists in gaining access to information on tourist locations by fetching information from the web, which also aids in the decision-making process for the tourists. It uses hybrid recommendation technique to implement a recommendation filtering system that can improve the efficiency and autonomy of the system in assisting tourist in decision making process based on user preferences.	The system has no provision of advertisement platform or content scheduler and unavailability of a fully functional reservation platform.
4.	A Review of Machine Learning (ML) Algorithms used for Modeling Travel Mode Choice	Juan D. Pineda-Jaramillo	In this paper different methodologies used by researchers for modelling travel mode choice are collected and compared with the Multinomial Logit Model (MNL). The application of different ML techniques in the field of transportation intends to meet the challenges of growing travel demands, safety concerns and energy consumption	The Multinomial Logit Model is compared with Artificial Neural Networks (ANN), Decision Trees (DT), Support Vector Machines (SVM) and Cluster Analysis (CA) based on various parameters and with different datasets. After the comparison of MNL with all the other Machine Learning algorithms, it was found that all the other algorithms that model travel mode choice	The paper does not explore incorporating the behavioural assumption that considers alternative-specific, which is enabled by the data framework of logit models “layered” into ML techniques and Deep Learning techniques for modelling travel mode choice.

				<p>like ANNs, DTs, SVMs and CA algorithms perform better than MNL for almost every case.</p> <p>Furthermore, it was also found that, if multiple Decision Trees are combined in a Random Forest algorithm, and used for the modelling, the outcomes are better than any other machine learning algorithm.</p>	
5.	Design and Development of Tour Management System using Android	Aishwarya Bhat, Ayesha Thasneema, Joylin Rosario and Judith Jennifer Rodrigues	This paper described about the creation and deployment of a tour management system using android. The system provides a good way for the users to plan their trips, since it provides the detailed information about the tourist places including description, image and map.	The system has five modules namely, user authentication, search destination, create tour, view order and give feedback. This application provides tourists with description, image, weather condition, maps of the places which guides the tourist. It also, provides best hotels, restaurants and travelling facility based on the user feedback. It provides distance between current and target location as well. It also provides customised packages, based on the cost and number of days	The system recommendation system is based only on other user reviews and does not take into account the user preference, cost and other factors.

2.2 Problem Definition

As we know that nowadays, with living standards getting better there is a huge craze of tourism and this is only increasing day by day in the whole world. With this increase in popularity there also comes various problems ranging from finding and searching for a place to travel to and then searching for the hotels, restaurants etc. to stay and visit to the tiresome and hectic manual booking procedure. Along with this, there are also other problems like less information about the availability of hotels and restaurants at the time of making the travel plan and of how to connect with the users to take their opinions and feedbacks of the services they used in order to provide other new users with another factor upon which they can base their choices and can get better options.

Hence, we came up with a Travel Management System that not only can be accessed and used from anywhere, at any time but also provides the user with services like user feedback on the places they visited and the facilities they used, an AI chatbot that can resolve any of the user queries in an interactive and easy to use manner, and recommendation system that takes the user preferences and other user reviews as parameters to recommend the best suited and available hotels, restaurants, etc. to make the decision making process of the user easier. The system is also made to provide a user-friendly environment with easy-to-use services and function.

3. Overview of the Work

3.1 Objectives of the Project

3.1.1 Product Perspective

Our product, Vacation Nation is a unique travel management system that traces its origin to already available travel management systems in the market, but with major improvements. It can be considered as a replacement for existing systems(products) which makes routine travel planning and scheduling with obsolete features. Major components of the product include the main user interface, recommendation system based on ML-model, User-reviews based selection model and multiple webpages incorporating basic functionalities and models.

3.1.2 Product Functions

Major functions of the product include:

- Presenting the user with the option to select the vacation destination of their choice and facilitate such options.
- Synchronize with the internal database and ML model to recommend destinations to uncertain

users based on their features such as age group, budget, trip type, etc.

- Present User with the option to give feedbacks about the travel agencies, destinations and trips, review and rate them.
- Present these feedbacks to the users who want to decide destinations based on such ratings.

3.1.3 User Classes and Characteristics

Our first set of users are frequent travellers, generally young-aged people who might need exclusive recommendations for travel destinations. They are very well acquainted with such exclusive features a product offers and might be the core userbase of our product.

Second set of users might be common people who travel annually or very rarely, specifically the middle-aged and old-aged people. These users go to religious places, places of historical importance or hill stations for their vacations and have general demands of a vacation trip. These users may or may not be interested in the exclusive services a product might offer but demand desirable and viable travel plan.

Another set of users might possibly be schools who take their students on an annual school trip or companies taking their employees on official visits to meeting destinations. This group is the least important to satisfy as their requirements aren't much than a cost-effective travel.

3.1.4 Operating Environment

The travel management recommendation system uses a web application so users require a hardware device to use webapp on like a desktop, laptop, tablet or a mobile phone. The software provided through web applications is developed for android, IOS, windows 7, windows 8, windows 10 operating systems. The travel management recommendation system requires use of Google Maps location APIs to get the location of destination.

3.1.5 Design and Implementation Constraints

Our product uses only 'English' language for its normal usage. This product can implement and process requests using only the above-mentioned language(s). Multiple language formats are at present not supported. Our product is at its pilot stage and has not been commercialized so the database needed for its commercialization is not yet supported and procured.

Additionally, such databases used in its nascent stage can have security issues. These can be overcome in later stages when the product has to be commercialized. Due to the nature of its working, it operates in a linear manner and doesn't support redundancy.

3.1.6 User Documentation

The user documentation will contain the following guidelines and manuals to help the user with smooth navigation and usage:

User manual – in accordance to Bureau of Indian Standards

Installation guide

Security Guidelines

3.1.7 Assumptions and Dependencies

It is assumed that the user enters required data in correct format without any bias when asked for their preferences or while reviewing the trips and destinations. Anomalies in the entered data could possibly affect the working of recommendation systems. Our product uses Google Maps APIs for locating the destinations and is dependent on its precision and working for smooth usage.

3.2 Software Requirements

3.2.1 System Feature 1

Chat Bot

This is the part of our project in which the chat bot will answer the customers queries 24/7 without any human affords. It can clarify the doubts of the customers and suggests the best booking options according to the user requirements and based on the previous users experience which was collected by the feedback form.

Medium Priority Feature as this feature benefits users and helps solve their queries but doesn't particularly affect the normal working or to the business.

3.2.2 System Feature 2

Feedback:

Providing good experience to the users is the main theme of any organization and the best way to make it more efficient is by collecting the users feedback and make use it as a suggestion and improve our system and also, we can make use this feedback as an input to give suggestions to the users based on ratings of the particular destination.

High Priority feature as this unique feature offers users to view the ratings and decide their destinations based on the ratings and reviews of other users. This helps users feel connected to and communicating to already experienced travelers and help make correct choices.

3.2.3 System Feature 3

User Friendly Options:

The Travel Management System is mainly dependent on its user's experience. The users can easily update their personal details and the can modify or delete their account and they can choose their requirements in an easy way. If any user faces any difficulties in booking their

rides, accommodation, etc. they will approach other TMS. To avoid losing business or customers, building a user-friendly booking tool is most important.

Low priority as it is mostly general and is always incorporated without any special mention.

3.2.4 System Feature 4

ML-based Recommendation Model:

This is the most important feature and is the most intriguing one as it recommends users the travel destinations based on their features. These features as of now include their age group, budget, travel type (solo, friends, family), travel preference (beaches, hill stations, religious pilgrimage, casual sightseeing, etc.) and can evolve with time as database gets updated and so do the developers as they get to know about the working and user class more closely.

ML models used classify the user groups based on these features and scores calculated statistically. These ML models might include Random Forest Classifier, AdaBoost Classifier, Stochastic Gradient Descent Classifier and CART Decision Tree and consists of Stacked Pipeline to classify in a more efficient manner. It might later be developed with the help of Deep Learning Modules to enhance its working, precision and recommendation.

3.2.5 Non-function Requirements

3.2.5.1 Performance Requirements

The website should be smooth and lag free from the perspective of the user. The website should run be able to run on all types of devices with different operating websites like Windows, Mac, Android and iOS and all types of screen sizes. The application should be scalable and should be able to handle multiple users at the same time without jeopardizing the functionality or the usability of any of the website.

3.2.5.2 Safety Requirements

The scope of different type of users should be clearly defined. Different users should have access to information which are relevant to them and not any unnecessary information nor should they be able to change information that is above their scope. The system should also be redundant to minimize the loss of data. The password and other critical PII (Personal Identification Information) should be kept safe and encrypted to prevent any losses and for the safety and security of the users.

3.2.5.3 Security Requirements

The scope of different type of users should be clearly defined. Different users should have access to information which are relevant to them and not any unnecessary information nor should they be able to change information that is above their scope. The system should also be redundant to minimize the loss of data. The password and other critical PII (Personal Identification Information) should be kept safe and encrypted to prevent any losses and for the safety and security of the users.

3.2.5.4 Software Quality Attributes

The website should be adaptable, available, correct, flexible, interoperable, maintainable, portable, reliable, reusable, robust, testable, and usable all the relevant parties.

3.2.5.5 Business Rules

The Administrators of the product have complete authority over the product, database and its security and are responsible solely. They enforce the laws and regulations and change them as required. The vendors are solely responsible for the end-delivery of the product to the users.

3.2.5.6 Other Requirements

Creating a database:

Tourism is currently recognized as a global industry that is growing at a high rate, like any other industry. This web-based application helps in maintaining the database. It has a friendly environment that connects customers willingly. Thus, it simplifies the process by saving time and efforts. It will help tour managers to control and handle the tour-related activities effectively and efficiently. A further modification could be possible where the system can be integrated with bigger organizations such as tourist agencies in order to help them.

3.3 Hardware Requirements

The travel Management recommendation system uses a web application, so users require a hardware device to use webapp on like a desktop, laptop, tablet or a mobile phone. The application is built to be used on windows 64-bit in case a user is using a Windows operating system, so the device must meet the requirements. Also, the setting up of an account is required to create travel plans using the device.

Requires GPU for computation of ML Models.

Requires Hosting platforms (for example, Heroku) to host and integrate modules.

4. System Design

4.1 Algorithms

4.1.1 Queries Chatbot

```
var coll = document.getElementsByClassName("collapsible");
```

```
function getTime()
```

```
// Gets the first message
```

```
function firstBotMessage() {
```

```

let firstMessage = "How's it going?"

document.getElementById

// Retrieves the response

function getHardResponse(userText)

// Handles sending text via button clicks

function buttonSendText(sampleText)

function getBotResponse(input) {

//use of switch case

switch(input.toLowerCase()){

case "hello":

return "Hello there! How can I help you?";

break;

```

4.1.2 Recommendation System

Importing Required Libraries

#Importing the dataset

```
df=pd.read_csv("recommend_user.csv")
```

#Label Encoding the non-numeric features

```
from sklearn.preprocessing import LabelEncoder
```

```
labelen = LabelEncoder()
```

#Splitting the dataset into

#features and class label

```
X=df.iloc[:,0:7]
```

```
y=df.iloc[:,7:8]
```

Training Random Forest Classifier

```
rf = RandomForestClassifier(random_state=1)
```

```
rf.fit(X,y)
```

Training AdaBoost Classifier

```
abc = AdaBoostClassifier(n_estimators=5)
```

```
abc.fit(X,y)
```

Training Stochastic Gradient Descent Classifier

```
sgd=SGDClassifier(loss='modified_huber',shuffle=True,random_state=1)
```

```
sgd.fit(X,y)
```

Training Stacking Classifier

```
from sklearn.ensemble import StackingClassifier
```

```
#Define the base models
```

```
level0 = list()
```

```
level0.append(('rf', RandomForestClassifier()))
```

```
level0.append(('abc', AdaBoostClassifier()))
```

```
level0.append(('sgd',SGDClassifier()))
```

```
#Define meta learner model
```

```
level1 = DecisionTreeClassifier()
```

```
#Define the stacking ensemble
```

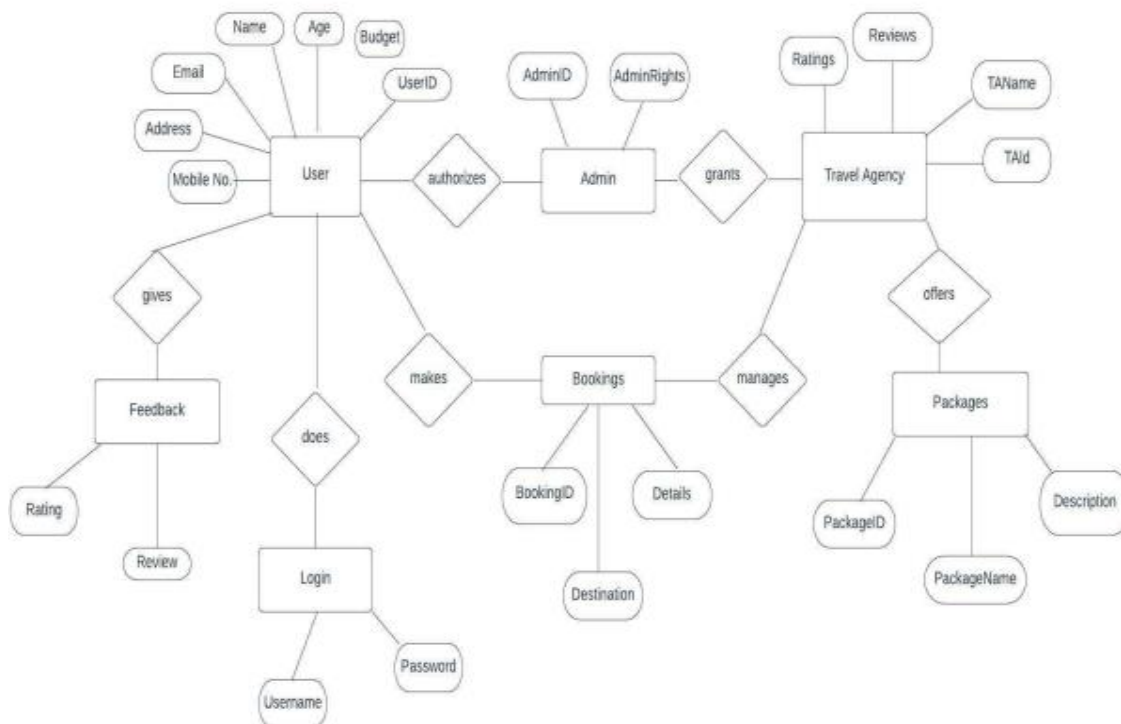
```
model = StackingClassifier(estimators=level0, final_estimator=level1, cv=2)
```

```
model.fit(X,y)
```

Deploy the 'model' for predictions.

4.2 System Modeling

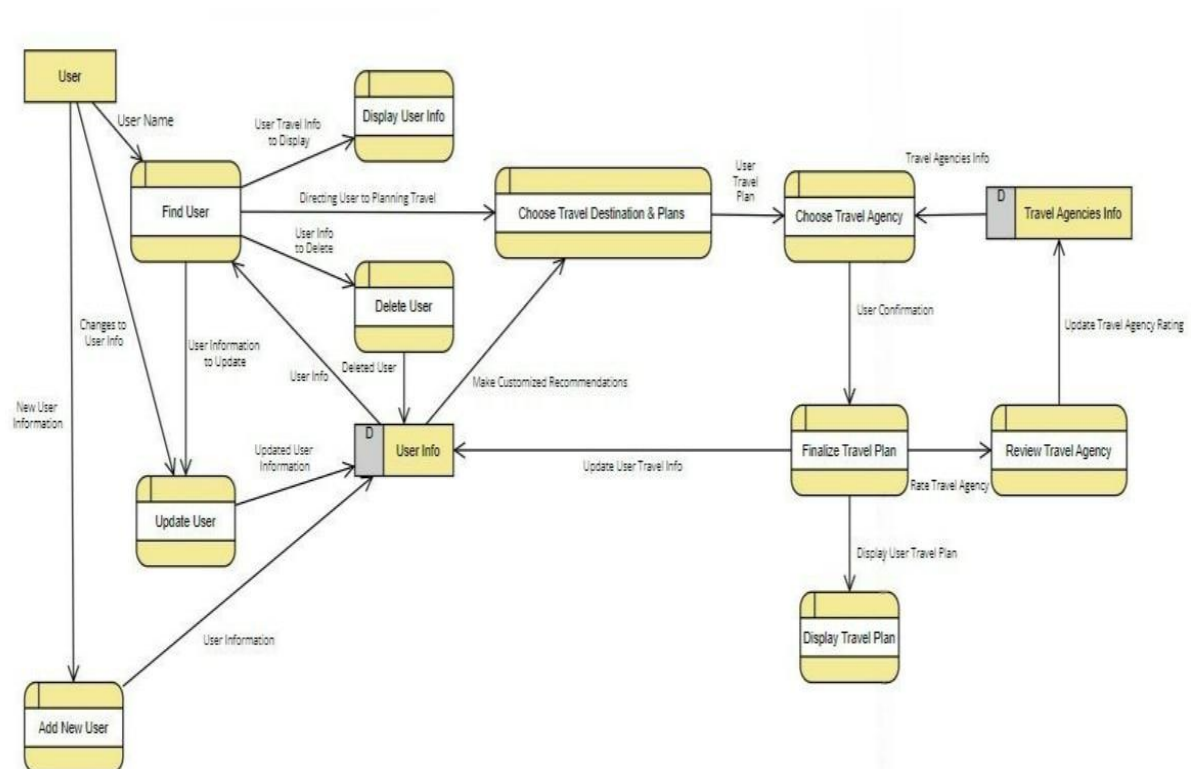
4.2.1 ER Diagram



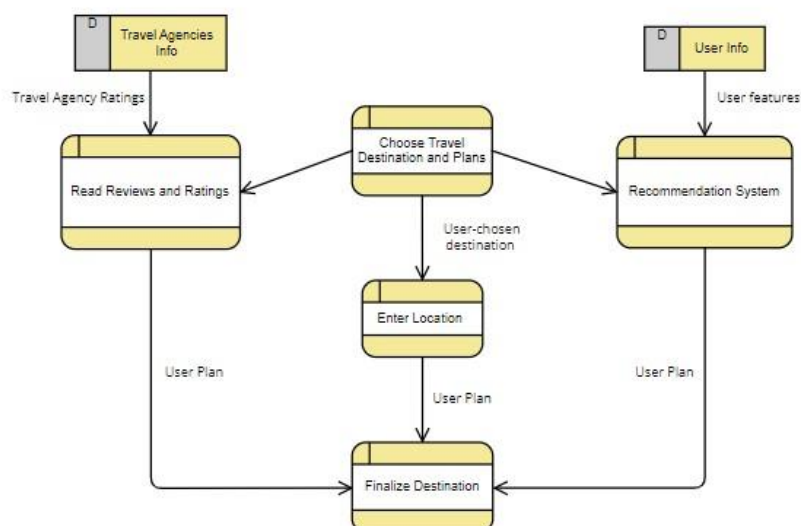
ER Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. They use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

4.2.2 Dataflow Diagram

Level 0:

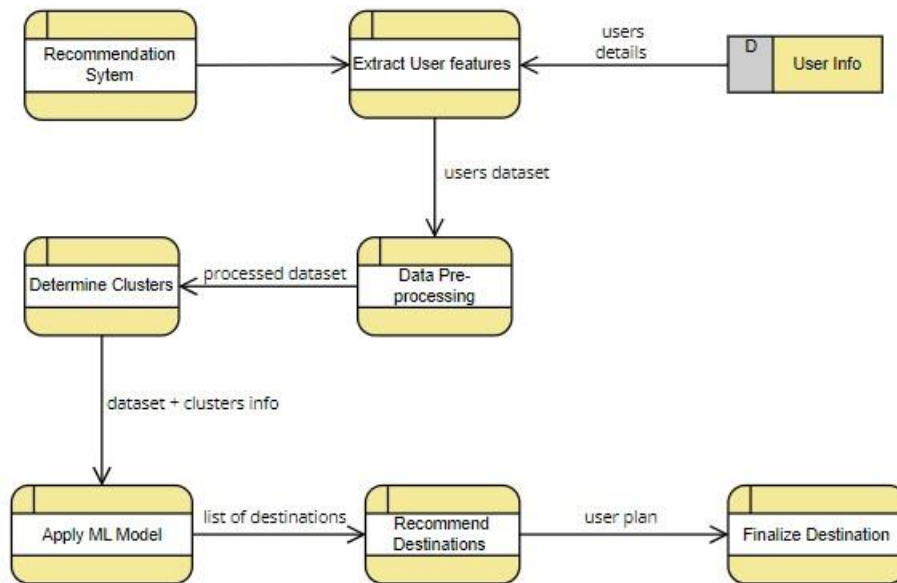


Level 1:



The Level 1 of DFD expands the dataflow structure while choosing travel destination and plans as illustrated in Level 0. It explains in details the various modes of execution.

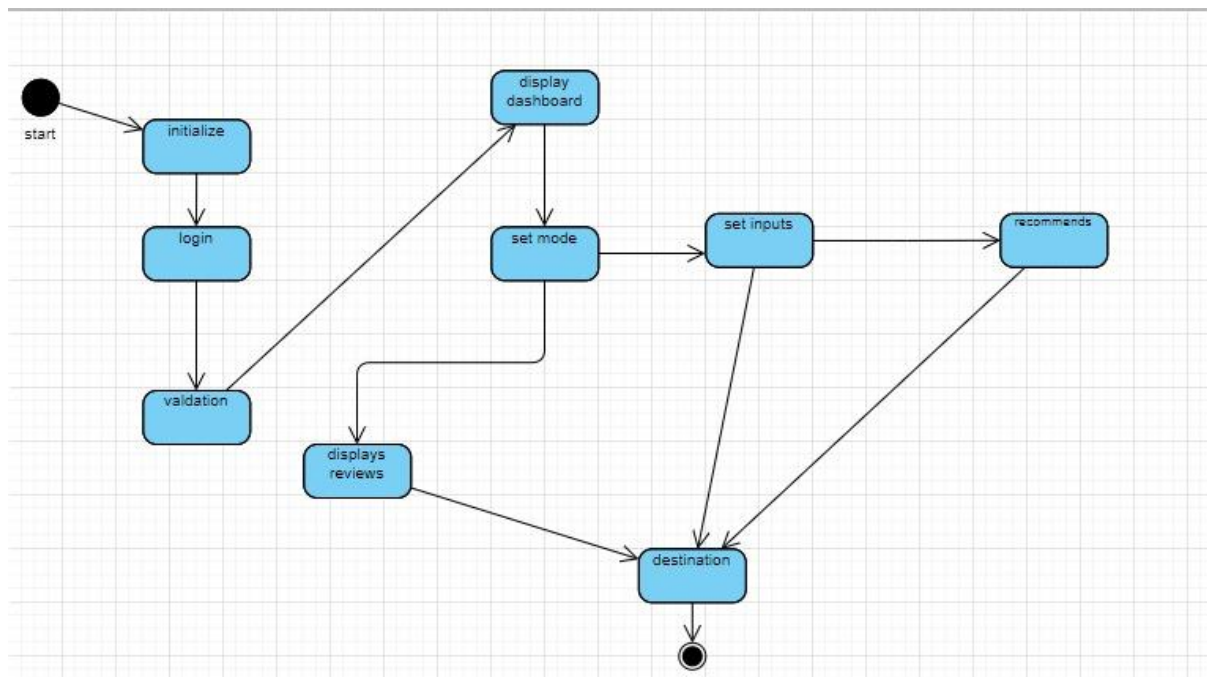
Level 2:



Level 2 of DFD expands the dataflow structure for the process of recommendation system as illustrated in Level 1. It explains in detail the processes and data flow involved in the execution of the ML model and recommendation.

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

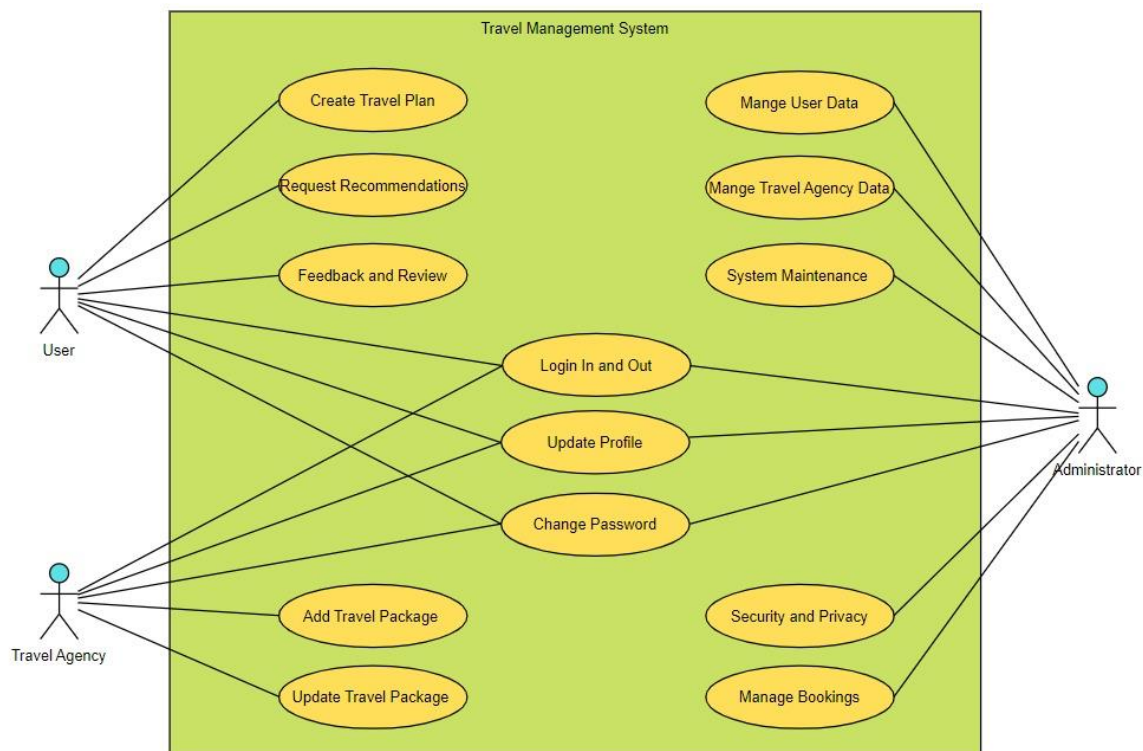
4.2.3 State Diagram



The state machine diagram shows the order of states underwent by an object within the system. It captures the software system's behaviour.

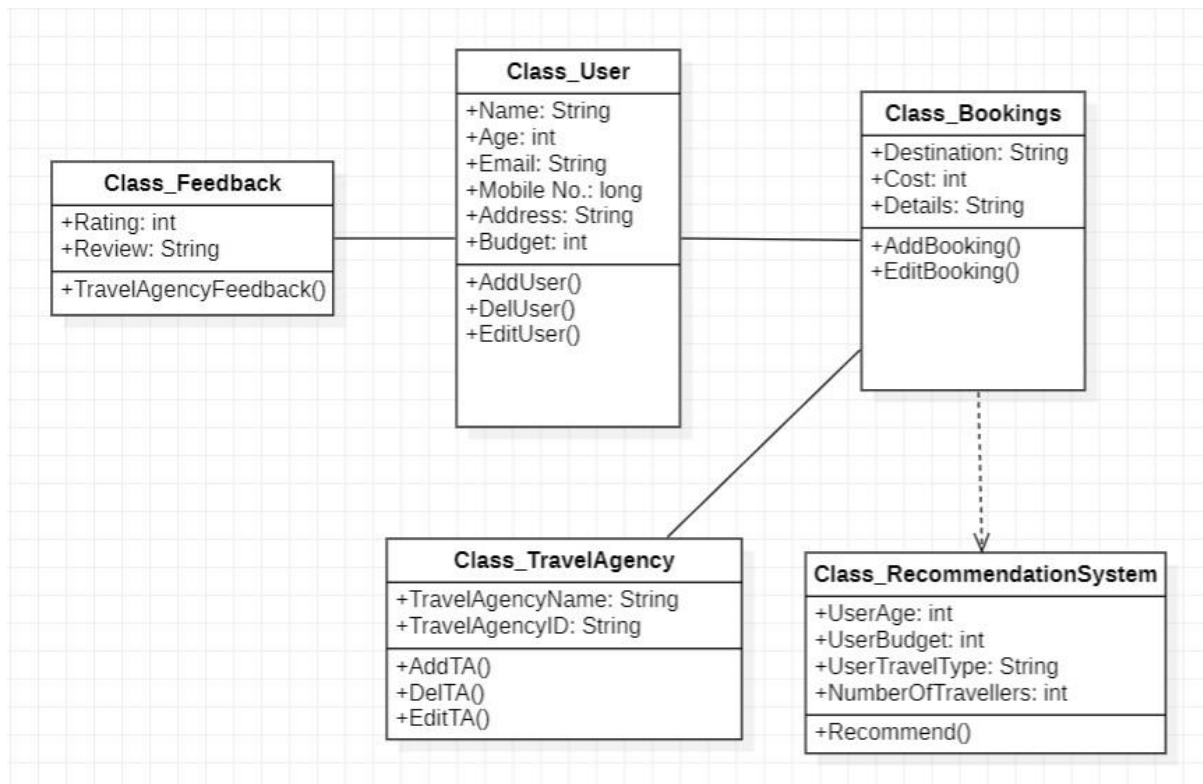
State machine diagram typically are used to describe state-dependent behaviour for an object. An object responds differently to the same event depending on what state it is in.

4.2.4 Use Case Diagram



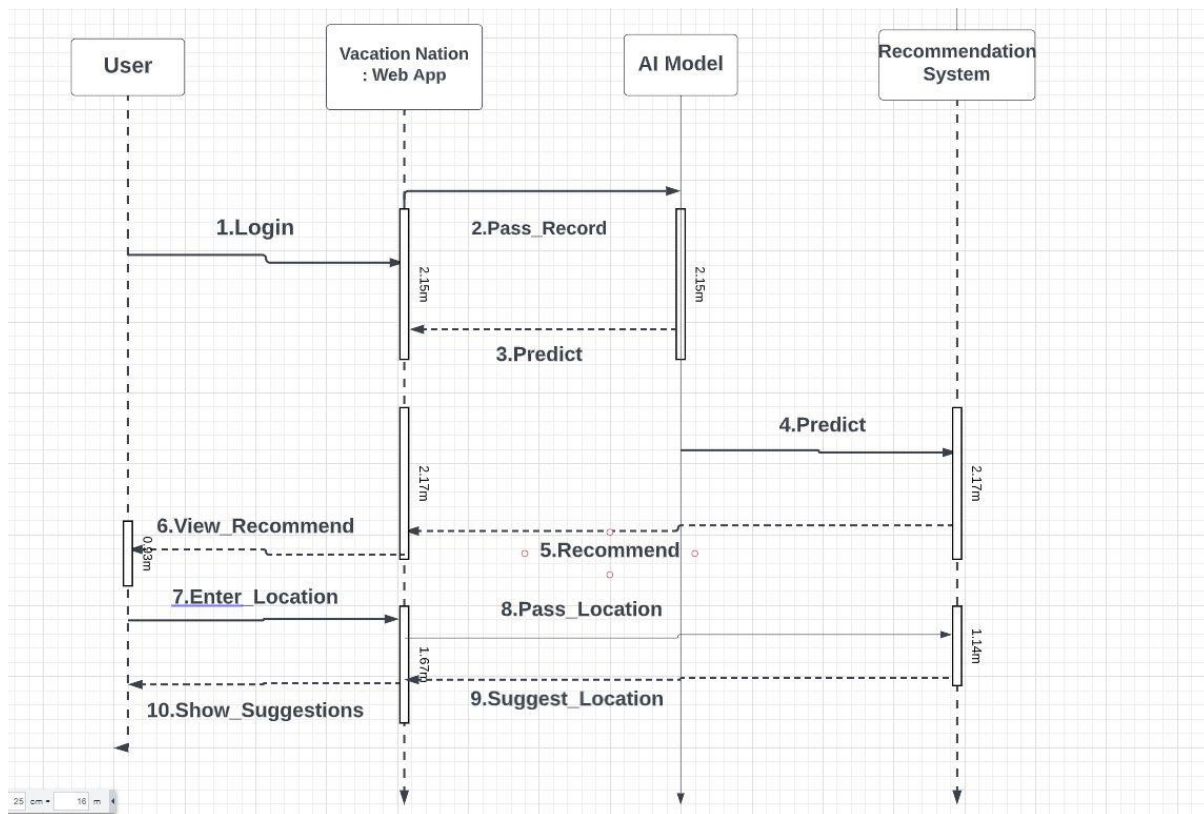
Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

4.2.5 Class Diagram



The class diagram is one of the types of UML diagrams which is used to represent the static diagram by mapping the structure of the systems using classes, attributes, relations, and operations between the various objects. A class diagram has various classes; each has three-part; the first partition contains a Class name which is the name of the class or entity which is participated in the activity, the Second partition contains class attributes that show the various properties of the class, the third partition contains class operations which shows various operations performed by the class, relationships show the relation between two classes.

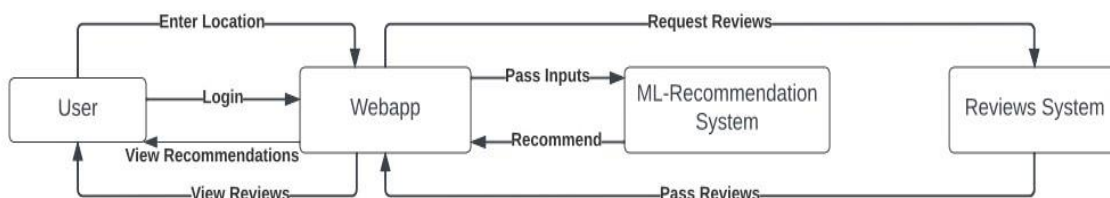
4.2.6 Sequence Diagram



A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together.

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios.

4.2.7 Collaboration Diagram



This mainly puts emphasis on the structural aspect of the interaction diagram, i.e., how lifelines are connected. While the sequence diagram focusses on the instance at which interactions happen, collaboration diagram focusses plainly on interconnections without instances as illustrated above.

5. Implementation

5.1 Description of Modules/Programs

Chatbot is the part of our project in which the chat bot will answer the customers queries 24/7 without any human affords. It can clarify the doubts of the customers and suggests the best booking options according to the user requirements and based on the previous users experience which was collected by the feedback form.

Providing good experience to the users is the main theme of any organization and the best way to make it more efficient is by collecting the users feedback and make use it as a suggestion and improve our system and also, we can make use this feedback as an input to give suggestions to the users based on ratings of the particular destination.

The Travel Management System is mainly dependent on its user's experience. The users can easily update their personal details and the can modify or delete their account and they can choose their requirements in an easy way. If any user faces any difficulties in booking their rides, accommodation, etc. they will approach other TMS. To avoid losing business or customers, building a user-friendly booking tool is most important.

ML-based Recommendation Model is the most important feature and is the most intriguing one as it recommends users the travel destinations based on their features. These features as of now include their age group, budget, travel type (solo, friends, family), travel preference (beaches, hill stations, religious pilgrimage, casual sightseeing, etc.) and can evolve with time as database gets updated and so do the developers as they get to know about the working and user class more closely.

ML models used classify the user groups based on these features and scores calculated statistically. These ML models might include Random Forest Classifier, AdaBoost Classifier, Stochastic Gradient Descent Classifier and CART Decision Tree and consists of Stacked Pipeline to classify in a more efficient manner. It might later be developed with the help of Deep Learning Modules to enhance its working, precision and recommendation.

5.2 Source Code

5.2.1 ML-model and Flask Integration

model.py

```
6. #Importing Required Libraries
7. import numpy as np
8. import pandas as pd
9. from sklearn.ensemble import RandomForestClassifier
10. from sklearn.ensemble import AdaBoostClassifier
11. from sklearn.linear_model import SGDClassifier
12. from sklearn.tree import DecisionTreeClassifier
13. import pickle
```

```

14.
15.#Importing the dataset
16.df=pd.read_csv("recommend_user.csv")
17.
18.#Label Encoding the non-numeric features
19.from sklearn.preprocessing import LabelEncoder
20.labelen = LabelEncoder()
21.df['Travel Group'] = labelen.fit_transform(df['Travel Group'])
22.df['Type Of Destination'] = labelen.fit_transform(df['Type Of
    Destination'])
23.df['Occasion'] = labelen.fit_transform(df['Occasion'])
24.df['Regions To Visit'] = labelen.fit_transform(df['Regions To Visit'])
25.
26.#Splitting the dataset into
27.#features and class label
28.X=df.iloc[:,0:7]
29.y=df.iloc[:,7:8]
30.y=y.values.flatten()
31.
32.rf = RandomForestClassifier(random_state=1)
33.rf.fit(X,y)
34.
35.abc = AdaBoostClassifier(n_estimators=5)
36.abc.fit(X,y)
37.
38.sgd=SGDClassifier(loss='modified_huber',shuffle=True,random_state=1,max
    _iter=50000)
39.sgd.fit(X,y)
40.
41.from sklearn.ensemble import StackingClassifier
42.#Define the base models
43.level0 = list()
44.level0.append(('rf', RandomForestClassifier()))
45.level0.append(('abc', AdaBoostClassifier()))
46.level0.append(('sgd',SGDClassifier()))
47.#Define meta learner model
48.level1 = DecisionTreeClassifier()
49.#Define the stacking ensemble
50.model = StackingClassifier(estimators=level0, final_estimator=level1,
    cv=5)
51.
52.model.fit(X,y)
53.pickle.dump(model,open('model.pkl','wb'))
54.

```

basics.py

```
from flask import Flask, render_template
```

```

app = Flask(__name__)

@app.route('/home')
def home():
    return render_template('recommend1.html')

if __name__ == "__main__":
    app.run(debug=True)

```

app.py

```

from flask import Flask, render_template, request
import pickle
import numpy as np

model = pickle.load(open('model.pkl', 'rb'))

app = Flask(__name__)

@app.route('/')
def man():
    return render_template('recommend1.html')

@app.route('/predict', methods=['POST'])
def home():
    data1 = request.form['min_age']
    data2 = request.form['max_age']
    data3 = request.form['budget']
    data4 = request.form['travelgroup']
    data5 = request.form['typeofdestination']
    data6 = request.form['occasion']
    data7 = request.form['placestovisit']
    arr = np.array([[data1, data2, data3, data4, data5, data6, data7]])
    pred = model.predict(arr)
    return render_template('recommend2.html', data=pred)

if __name__ == "__main__":
    app.run(debug=True)

```

5.2.2 Chatbot Model

// Collapsible

```
var coll = document.getElementsByClassName("collapsible");
```

```

for (let i = 0; i < coll.length; i++) {
  coll[i].addEventListener("click", function () {
    this.classList.toggle("active");
    var content = this.nextElementSibling;
    if (content.style.maxHeight) {
      content.style.maxHeight = null;
    } else {
      content.style.maxHeight = content.scrollHeight + "px";
    }
  });
}

```

```

function getTime() {
  let today = new Date();
  hours = today.getHours();
  minutes = today.getMinutes();
  if (hours < 10) {
    hours = "0" + hours;
  }
  if (minutes < 10) {
    minutes = "0" + minutes;
  }
  let time = hours + ":" + minutes;
  return time;
}

```

// Gets the first message

```

function firstBotMessage() {
  let firstMessage = "How's it going?"

```



```

    document.getElementById("botStarterMessage").innerHTML = '<p
class="botText"><span>' + firstMessage + '</span></p>';

    let time = getTime();

    $("#chat-timestamp").append(time);

    document.getElementById("userInput").scrollIntoView(false);
}
firstBotMessage();
// Retrieves the response
function getHardResponse(userText) {
    let botResponse = getBotResponse(userText);

    let botHtml = '<p class="botText"><span>' + botResponse + '</span></p>';

    $("#chatbox").append(botHtml);

    document.getElementById("chat-bar-bottom").scrollIntoView(true);
}

//Gets the text text from the input box and processes it
function getResponse() {
    let userText = $("#textInput").val();

    if (userText == "") {
        userText = "I love Code Palace!";
    }

    let userHtml = '<p class="userText"><span>' + userText + '</span></p>';

    $("#textInput").val("");

    $("#chatbox").append(userHtml);

    document.getElementById("chat-bar-bottom").scrollIntoView(true);

    setTimeout(() => {
        getHardResponse(userText);
    }, 1000);
}

```

```

    }, 1000)

}

// Handles sending text via button clicks
function buttonSendText(sampleText) {
    let userHtml = '<p class="userText"><span>' + sampleText + '</span></p>';
    $("#textInput").val("");
    $("#chatbox").append(userHtml);
    document.getElementById("chat-bar-bottom").scrollIntoView(true);
}

function sendButton() {
    getResponse();
}

function heartButton() {
    buttonSendText("Heart clicked!")
}

// Press enter to send a message
$("#textInput").keypress(function (e) {
    if (e.which == 13) {
        getResponse();
    }
});

```

5.2.3 User-friendly UI

```

<!DOCTYPE html>
<html>

<head>
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">

```

```

    <link type="text/css" rel="stylesheet" href="../index-
style/materialize.min.css" media="screen,projection" />
    <link type="text/css" rel="stylesheet" href="../index-style/main.css" />
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.1.0/css/all.css"
integrity="sha384-
lKuwvrZot6UHsBSfcMv0kWWlCMgc0TaWr+30HWe3a4ltaBwTZhyTEggF5tJv8tbt"
crossorigin="anonymous">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Welcome to Vacation Nation</title>
</head>

<body class="scrollspy">
    <a name="home"></a>
    <div class="navbar-fixed">
        <nav class="green">
            <div class="container">
                <a href="#home" class="brand-logo">Vacation Nation</a>
                <a href="#" data-activates="mobile-nav" class="button-
collapse">

                    <i class="material-icons">menu</i>
                </a>

                <ul class="right hide-on-med-and-down">

                    <li>
                        <a href="#popular">Popular Places</a>
                    </li>
                    <li>
                        <a href="gallery.html">Gallery</a>
                    </li>
                    <li><a class="dropdown-button" href="#" data-
activates="dropdown4">Places<i class="material-icons
right">arrow_drop_down</i></a>
                    </li>

                    <li><a class="dropdown-button" href="#" data-
activates="dropdown3">More<i class="material-icons
right">arrow_drop_down</i></a>
                    </li>
                    <li>
                        <a href="aichatbot/chatbot.html" >Chatbot</a>
                    </li>
                    <li>
                        <a href="../index.html" onclick="logout()">logout</a>
                    </li>
                </ul>
            </div>
        </nav>
    </div>

```

```

        </div>
    </nav>
    <ul id="dropdown1" class="dropdown-content">
        <li><a href="setting.html">Settings</a></li>
        <li class="divider"></li>
        <li><a href="about.html">About Us</a></li>
        <li class="divider"></li>
        <li><a href="team.html">Our Team</a></li>
        <li class="divider"></li>
        <li><a href="contact.html">Contact</a></li>
    </ul>
    <ul id="dropdown3" class="dropdown-content">
        <li><a href="setting.html">Settings</a></li>
        <li class="divider"></li>
        <li><a href="about.html">About Us</a></li>
        <li class="divider"></li>
        <li><a href="team.html">Our Team</a></li>
        <li class="divider"></li>
        <li><a href="contact.html">Contact</a></li>
    </ul>
    <ul id="dropdown2" class="dropdown-content">
        <li><a href="type/beaches.html">Beaches</a></li>
        <li class="divider"></li>
        <li><a href="type/hill-station.html">Hill-Stations</a></li>
        <li class="divider"></li>
        <li><a href="type/historical.html">Historical Heritage</a></li>
        <li class="divider"></li>
        <li><a href="type/parks.html">National Parks</a></li>
        <li class="divider"></li>
        <li><a href="type/religious.html ">Religious Richness</a></li>
        <li class="divider"></li>
        <li><a href="type/honeymoon.html ">Honeymoon Destinations</a></li>
        <li class="divider"></li>
    </ul>
    <ul id="dropdown4" class="dropdown-content">
        <li><a href="type/beaches.html">Beaches</a></li>
        <li class="divider"></li>
        <li><a href="type/hill-station.html">Hill-Stations</a></li>
        <li class="divider"></li>
        <li><a href="type/historical.html">Historical Heritage</a></li>
        <li class="divider"></li>
        <li><a href="type/parks.html">National Parks</a></li>
        <li class="divider"></li>
        <li><a href="type/religious.html ">Religious Richness</a></li>
        <li class="divider"></li>
        <li><a href="type/honeymoon.html ">Honeymoon Destinations</a></li>
        <li class="divider"></li>
    </ul>

```

```

</div>
<!-- we put the sidebar li elements outside the div because its fixed div
and we wont be able to access these links -->
<ul class="side-nav" id="mobile-nav">
  <li>
    <a href="#popular">Popular Places</a>
  </li>
  <li>
    <a href="gallery.html">Gallery</a>
  </li>
  <li><a class="dropdown-button" href="#" data-
activates="dropdown2">Places<i class="material-icons right">filter_hdr</i></a>
  </li>
  <li><a class="dropdown-button" href="#" data-
activates="dropdown1">More<i class="material-icons
right">arrow_drop_down</i></a>
  </li>
  <li>
    <a href="../index.html" onclick="logout()">logout</a>
  </li>
</ul>

<!-- Section: Slider -->
<section class="slider ">
  <ul class="slides ">
    <li>
      
      <div class="caption center-align ">
        <h2>Take Your Dream Vacation</h2>
        <h5 class="light grey-text text-lighten-3 hide-on-small-
only ">Adventure awaits, go find it.</h5>
        <a href="contact.html " class="btn btn-large green
">Contact Us</a>
      </div>
    </li>
    <li>
      
      <div class="caption left-align ">
        <h2>We Work With All Budgets</h2>
        <h5 class="light grey-text text-lighten-3 hide-on-small-
only ">Book a ticket and just leave.</h5>
        <a href="contact.html " class="btn btn-large green
">Contact Us</a>
      </div>
    </li>
    <li>
      
      <div class="caption right-align ">

```

```

        <h2>Group & Individual Gataways</h2>
        <h5 class="light grey-text text-lighten-3 hide-on-small-only ">Experience. Tourism. These are as education in themselves.</h5>
        <a href="contact.html " class="btn btn-large green
">Contact Us</a>
    </div>
</li>
</ul>
</section>

<!-- Section: Search -->

<section id="search" class="section section-search green darken-1 white-text center scrollspy ">
    <div class="container">
        <div class="row">
            <div class="col s12">
                <h3>Search Destinations </h3>
                <div class="input-field ">
                    <input type="text" class="white grey-text
autocomplete" placeholder="Agra,Manali,Gulmohar etc..." id="autocomplete-input
">
                </div>
            </div>
        </div>
    </div>
</section>

<!-- Section: Icon Boxes -->
<section class="section section-icons grey lighten-4 center ">
    <div class="container ">
        <div class="row ">
            <div class="col s12 m4 ">
                <div class="card-panel ">
                    <i class="material-icons large green-text ">room</i>
                    <h4><a href="setting.html">Pick Destinations</a></h4>
                    <p>Feed your wanderlust.</p>
                </div>
            </div>
            <div class="col s12 m4">
                <div class="card-panel">
                    <i class="material-icons large green-text">store</i>
                    <h4><a href="review.html">Pick from Reviews</a></h4>
                    <p>See what others say.</p>
                </div>
            </div>
            <div class="col s12 m4">
                <div class="card-panel ">

```

```

        <a href="recommend1.html"></a>
        <i class="material-icons large green-text
">airplanemode_active</i>
        <h4><a href="recommend1.html">Recommendation
System</a></h4>
        <p>No idea? Let us tell you</p>
    </div>
</div>
</div>
</div>
</section>

<!-- Section: Popular -->

<section id="popular" class="section section-popular scrollspy ">
    <div class="container ">
        <div class="row ">
            <h4 class="center ">
                <span class="green-text ">Popular</span> Places</h4>
            <div class="col s12 m4 ">
                <div class="card ">
                    <div class="card-image ">
                        
                    <span class="card-title ">GOA</span>
                </div>
                <div class="card-content ">
                    <p>Sandy beaches, warm sunsets, beautiful villages
and an atmosphere filled with joy .</p>
                </div>
            </div>
            <div class="col s12 m4 ">
                <div class="card ">
                    <div class="card-image ">
                        
                    <span class="card-title ">Rameshwaram</span>
                </div>
                <div class="card-content ">
                    <p>Rameshwaram is one of the holiest places in
India and is located on a beautiful island. </p>
                </div>
            </div>
            <div class="col s12 m4 ">
                <div class="card ">
                    <div class="card-image ">

```

```


    <span class="card-title ">Manali</span>
</div>
<div class="card-content ">
    <p>Manali is seated comfortably in the River Beas
Valley, Himachal Pradesh.</p>
</div>
</div>
</div>
<div class="col s12 m4 ">
    <div class="card ">
        <div class="card-image ">
            
            <span class="card-title ">Varanasi</span>
        </div>
        <div class="card-content ">
            <p>Varanasi, popularly known as the City of Lights
or Kashi, is a sacred city on the west bank of the River Ganga.</p>
        </div>
    </div>
</div>
<div class="col s12 m4 ">
    <div class="card ">
        <div class="card-image ">
            
            <span class="card-title ">Leh Ladakh</span>
        </div>
        <div class="card-content ">
            <p>Ladakh is an adventure playground for climbing,
jeeps tours, rafting and high-altitude trekking. </p>
        </div>
    </div>
</div>
<div class="col s12 m4 ">
    <div class="card ">
        <div class="card-image ">
            
            <span class="card-title ">Ooty</span>
        </div>
        <div class="card-content ">
            <p>Once regarded as the summer headquarters of the
East India Company, the Queen of the hills.</p>
        </div>
    </div>

```



```

        </div>
    </div>
</div>
<div class="row ">
    <div class="col s12 center ">
        <a href="contact.html " class="btn btn-large blue darken-3
">
            <i class="material-icons left ">send</i>Contact for
Booking
            </a>
        </div>
    </div>
</div>
</section>

<!-- Section: Follow -->

<section class="section section-follow green darken-2 white-text center ">
    <div class="container ">
        <div class="row ">
            <div class="col s12 ">
                <h4>Follow Vacation Nation</h4>
                <p>Follow us on social media for special offers.</p>
                <a href="https://facebook.com " target="_blank "
class="white-text ">
                    <i class="fab fa-facebook fa-4x "></i>
                </a>
                <a href="https://twitter.com " target="_blank "
class="white-text ">
                    <i class="fab fa-twitter fa-4x "></i>
                </a>
                <a href="https://linkedin.com " target="_blank "
class="white-text ">
                    <i class="fab fa-linkedin fa-4x "></i>
                </a>
                <a href="https://googleplus.com " target="_blank "
class="white-text ">
                    <i class="fab fa-google-plus fa-4x "></i>
                </a>
                <a href="https://pinterest.com " target="_blank "
class="white-text ">
                    <i class="fab fa-pinterest fa-4x "></i>
                </a>
            </div>
        </div>
    </div>
</section>

```

```

<!-- Section: Gallery -->

<section id="gallery" class="section section-gallery scrollspy ">
  <div class="container">
    <h4 class="center">
      <span class="green-text">Photo</span> Gallery
    </h4>
    <div class="row ">
      <div class="col s12 m3 ">
        
      </div>
      <div class="col s12 m3 ">
        
      </div>
      <div class="col s12 m3 ">
        
      </div>
      <div class="col s12 m3 ">
        
      </div>
    </div>
    <div class="row ">
      <div class="col s12 m3 ">
        
      </div>
      <div class="col s12 m3 ">
        
      </div>
      <div class="col s12 m3 ">
        
      </div>
      <div class="col s12 m3 ">
        
      </div>
    </div>
    <div class="row ">
      <div class="col s12 m3 ">
        

```

```

        </div>
        <div class="col s12 m3 ">
            
        </div>
        <div class="col s12 m3 ">
            
        </div>
        <div class="col s12 m3 ">
            
        </div>
    </div>
</div>
</section>

<!-- Footer section -->
<footer class="section green darken-1 white-text center ">
    <p class="flow-text ">Vacation Nation &copy; 2022</p>
</footer>

<!--Import jQuery before materialize.js-->
<script type="text/javascript " src="https://code.jquery.com/jquery-
3.2.1.min.js "></script>
<script type="text/javascript " src="..\index-style\materialize.min.js
"></script>
<script>
    $(document).ready(function() {
        // INIT sidebar
        $(".button-collapse ").sidebar();

        //INIT SLIDER
        $(".slider ").slider({
            indicators: false,
            height: 500,
            transition: 500,
            interval: 6000
        });

        //AUTOCOMPLETE
        $(".autocomplete ").autocomplete({
            data: {
                "Agra ": null,
                "gulmohar ": null,
                "Hawaii ": null,
                "Florida ": null,
                "California ": null,
            }
        });
    });

```

```

        "Jamaica ": null,
        "Europe ": null,
        "The Bahamas ": null
    }
});

//INIT SCROLLSPY
$(".scrollspy ").scrollSpy();

//init dropdown

$(".dropdown-button ").dropdown();

});

function logout() {
    alert("you are about to logout. Click OK to continue. ");
}
</script>
</body>
</html>

```

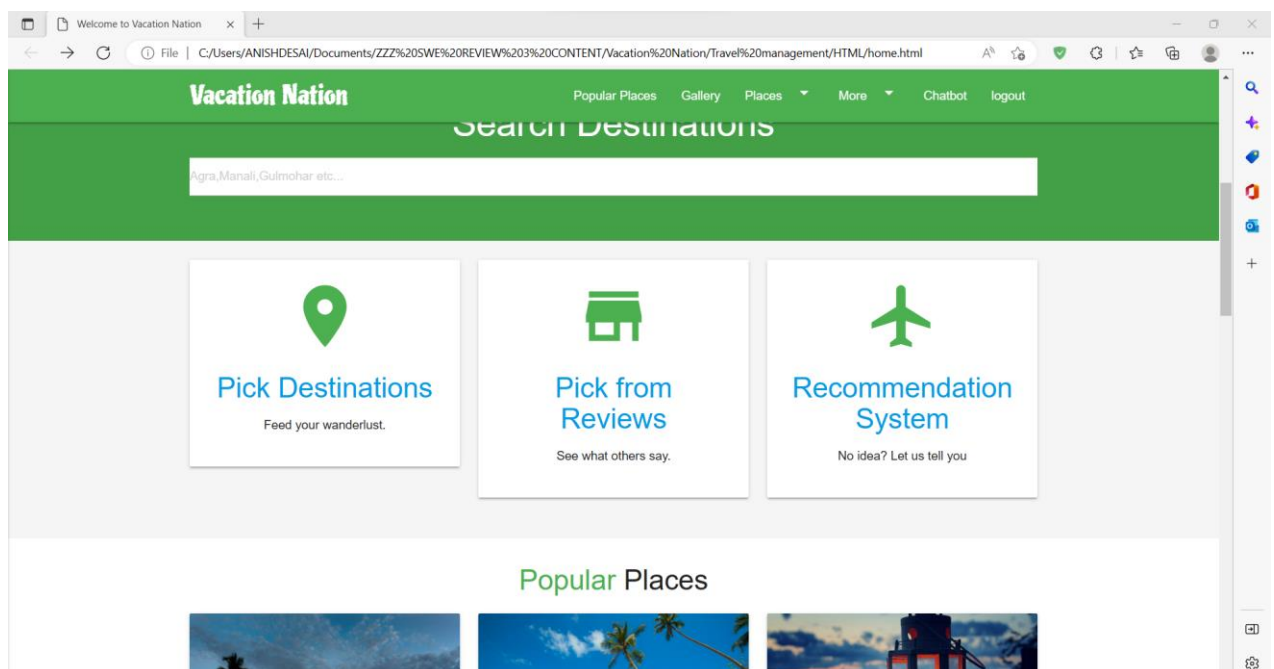
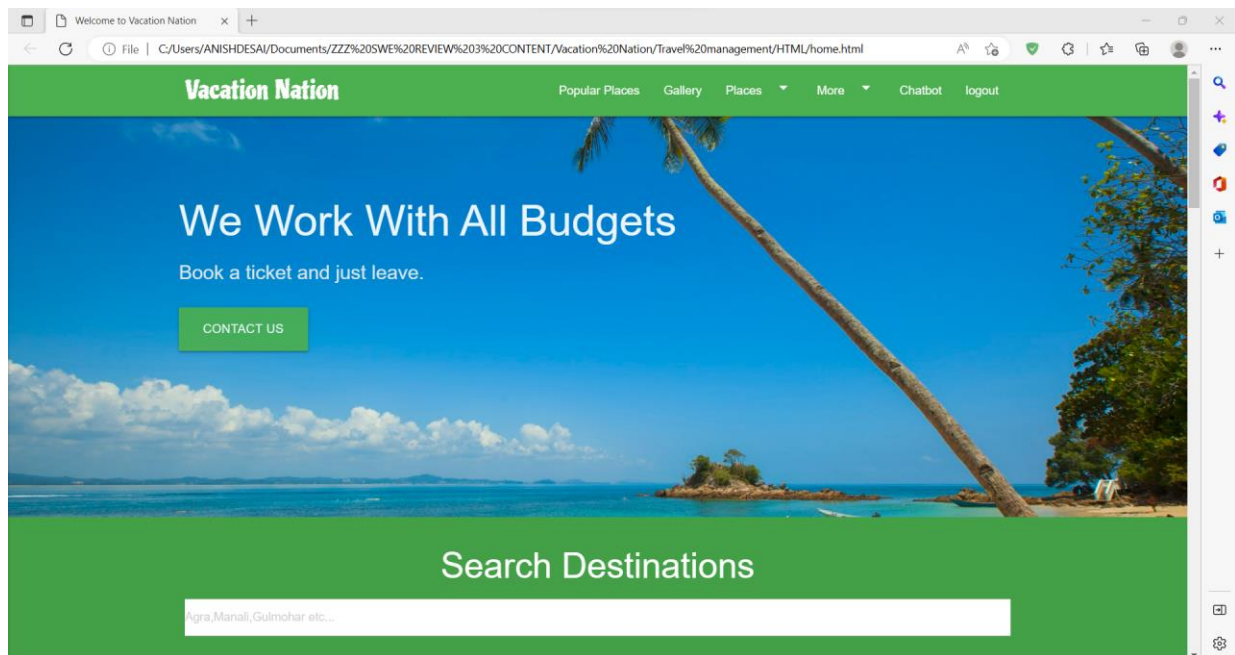
5.3 Test Cases

The test cases for the travel management recommendation include:

1. When the user has filled the details for the recommendation system parameters incorrectly, i.e., filled age as zero or negative.
2. If the user gives the budget of the trip as zero to the recommendation system. The budget allotted by the user must be >1000.
3. If the user searches for the destination in his mind with wrong details.
4. While using the AI chatbot if the user asks queries in a language other than English or enters numbers or symbols in the chatbot.
5. User enters the number of people in the trip as zero or negative.

6. Output and Performance Analysis

6.1 Execution Snapshots




hill-stations in India

Best Hill Stations in India


India has abundant hill stations spread all over the country that makes for a tranquil and picturesque destination for tourists. Kashmir has some of the most beautiful hill stations in India which include Pahalgam, Gulmarg, Srinagar and Sonmarg. In Himachal Pradesh also tourists can be spoilt for choices which include Shimla, Manali and Dalhousie. Mussoorie and Nainital are among the most popular hill stations in north India. Coorg, Ooty and Kodakanal are the most popular ones in South India. Matheran, Almora, McLeodganj, Gangtok, and Coonoor among others are the lesser crowded hill station in India.

Select your Dream Destination.




Leh Ladakh, Jammu & Kashmir
India's Own Moonland ★★★★★

Ladakh is a union territory in the Kashmir region of India. Formerly falling in the state of Jammu & Kashmir. Ladakh was administered a union territory on



Ooty, Tamil Nadu
Queen of the Nilgiris ★★★★★

Nestled amidst Nilgiri hills, Ooty, also known as Udagamandalam, is a hill station in Tamil Nadu which serves as a top-rated tourist destination. Once



Manali, Himachal Pradesh
India's Honeymoon capital ★★★★★

Nestled in between the snow-capped slopes of the Pir Panjal and the Dhauladhar ranges, Manali is one of the most popular hill stations in the country. With law-

recommend1.html

Vacation Nation

Trip Information

Age Range - Minimum Age

Age Range - Maximum Age

Budget

Travel Group ☒ Solo ☐ Family ☐ Friends ☐ Colleagues

Type Of Destination ☒ Hill Station ☐ Historical Heritage ☐ National Park ☐ Religious Richness ☐ Honeymoon Destination

Occasion ☐ Summer Holidays ☒ Casual Trip ☐ Organizational Trip

Regions To Visit

☐ North-Eastern India

☒ Northern India

☐ Southern India


☐ Western India

☐ Eastern India

[RECOMMEND](#)

Recommended Destination


Select your Dream Destination.



Leh Ladakh, Jammu & Kashmir
India's Own Moonland ★★★★★

Ladakh is a union territory in the Kashmir region of India. Formerly falling in the state of Jammu & Kashmir, Ladakh was administered a union territory on 31st October 2019. Extending from the Siachen Glacier to the main Great Himalayas, Ladakh is a land like no other. Ladakh is an adventure playground for climbing, jeeps tours, rafting and high-altitude trekking. Note that Leh Ladakh is inaccessible by road outside the summer months. The road pas.....


[Book Now!](#) [More Info](#)



Manali, Himachal Pradesh
India's Honeymoon capital ★★★★★★

Nestled in between the snow-capped slopes of the Pir Panjal and the Dhauladhar ranges, Manali is one of the most popular hill stations in the country. With jaw-dropping views, lush green forests, sprawling meadows carpeted with flowers, gushing blue streams, a perpetual fairy-tale like mist lingering in the air, and a persistent fragrance of pines - Manali has been blessed with extraordinary scenic beauty. From museums to temples, from quaint litt.....

[Book Now!](#) [More Info](#)



Kasol, Himachal Pradesh
Land of His Holiness ★★★★★★

Kasol is a quaint little village in Himachal situated along the banks of the river Parvati. Commonly known as the 'Amsterdam of India', Kasol is a tourist attraction that is rapidly gaining fame as a popular hub for trekkers, backpackers, and nature lovers. Kasol is situated 23 kms from Bhuntar and next to the religious town of Manikaran and is one of the best places in the country just to sit back and chill in the panorama of snow-clad mountains,

[Book Now!](#) [More Info](#)

Chat with us! 🗨️

18:01

How's it going?

Heart clicked!

hello

Hello there! How can I help you?

can you show the list of active packages?

Sure! 1.Shimla 2.Jaipur 3.Banglore

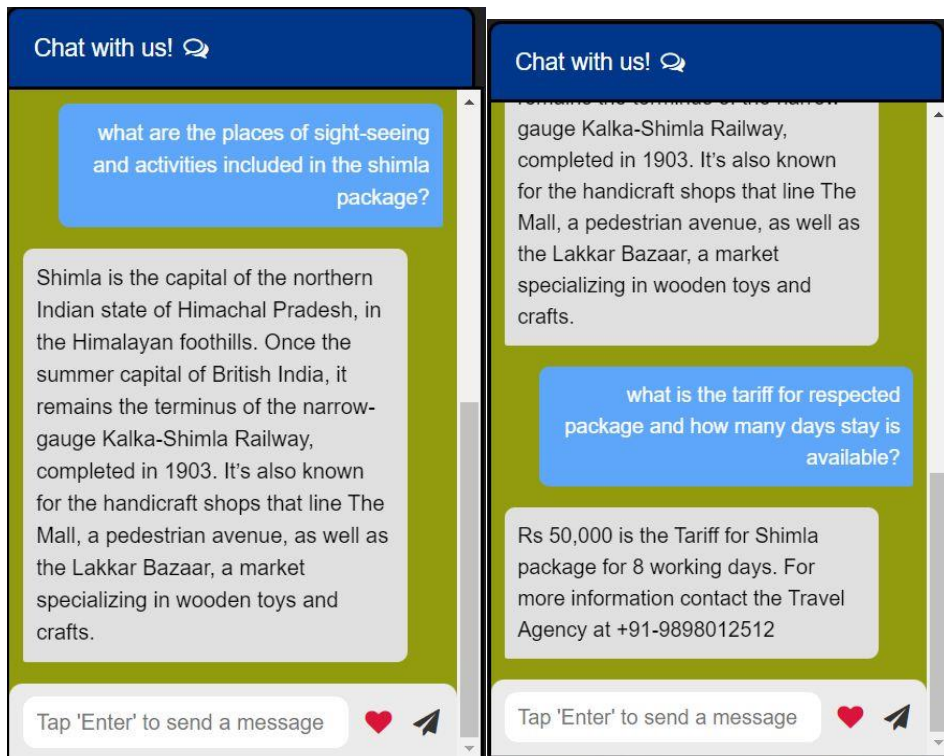
Tap 'Enter' to send a message ❤️ ➡️

Chat with us! 🗨️

18:01

How's it going?

Tap 'Enter' to send a message ❤️ ➡️



```

from sklearn.ensemble import StackingClassifier
# Define the base models
level0 = list()
level0.append(('rf', RandomForestClassifier()))
level0.append(('abc', AdaBoostClassifier()))
level0.append(('sgd', SGDClassifier()))
# Define meta learner model
level1 = DecisionTreeClassifier()
# Define the stacking ensemble
model = StackingClassifier(estimators=level0, final_estimator=level1, cv=2)

model.fit(X,y)
pred_model = model.predict([["20","20","0000","1","1","0","1"]])
y_pred_model

C:\Users\ANISHDESAI\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the
shape of y to (n_samples, ), for example using ravel().
    return f(*args, **kwargs)
C:\Users\ANISHDESAI\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: FutureWarning: Arrays of bytes/strings is being converted to decimal numbers if dtype='numeric'. This
behavior is deprecated in 0.24 and will be removed in 1.1 (renaming of 0.26). Please convert your data to numeric values explicitly instead.
    return f(*args, **kwargs)

array([1], dtype=int64)

```

6.2. Output – in terms of performance metrics

Lines Of Code (LOC) : 482*

Comprises of ML Model and ChatBot which makes the Main Model

* Excludes Client-side and Server-side scripting

Code Complexity :

Random Forest Classifier : $O(v * \log(n))$ [1]

AdaBoost Classifier : $O(n)$ [2]

Stochastic Gradient Descent Classifier : $O(n)$ [3]

CART Decision Tree Classifier : $O(v * \log(n))$ [4]

Overall Expected Code Complexity : $O(v * \log(n))$ where n is the number of records and v is the number of variables/attributes.

Customer Satisfaction

Measured in terms of APDEX score or Application Performance Index. It works by specifying a goal for how long a specific web request or transaction should take.

Average Response Time of our Model as measured

Execution time: 2.748255729675293 seconds

Accuracy

```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: %.2f%%" % (accuracy * 100.0))
```

Accuracy: 100.00%

Accuracy is defined as the proportion of correct predictions in all predictions made.

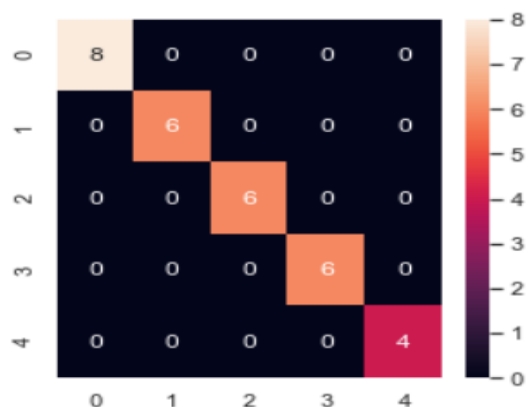
A high-range accuracy on account of custom-made dataset.

StackingClassifier() provides high-range accuracies above 99.8% consistently for major classification systems.

Confusion Matrix

```
import seaborn as sns
from sklearn.metrics import confusion_matrix
sns.set(rc={'figure.figsize':(4,4)})
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,annot=True)
```

<AxesSubplot:>



Matrix Plot of Actual Classification vs Predicted Classification.

Classification Report

```
#Performance Metrics of Model
from sklearn.metrics import classification_report
print('Classification Report : \n')
print(classification_report(y_test,y_pred))
```

Classification Report :

	precision	recall	f1-score	support
1	1.00	1.00	1.00	8
2	1.00	1.00	1.00	6
3	1.00	1.00	1.00	6
4	1.00	1.00	1.00	6
5	1.00	1.00	1.00	4
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

6.3 Performance comparison with existing works

The existing works either have very accuracy rates of 80%~87% or are not using such ML-based recommendation engines. Those using it take into consideration very few features which doesn't suit for real-time purposes.

7. Conclusion and Future Directions

This web application has been successfully created and saved all travel admin tour packages, booking, creation management and tour details in the database using this application. The application has been very well tested and bugs have been properly debugged. The testing also concluded that the system performance is satisfactory. All necessary output is generated. This system thus provides an easy way to automate all consumption functions. If this application is implemented with low consumption, it will be useful. The project can be further improved to make the website work in a very attractive and useful way than the current one. It was concluded that the application works well and meets the needs. The application is very well tested and bugs are properly debugged. It also acts as file sharing with valuable resources.

The tools used to research and book these experiences are constantly changing thanks to innovations in technology. With the development of OTAs and emerging and disruptive technologies, the travel landscape is constantly changing. The travel industry is forced to innovate at a surprising rate.

We are further going to work on the security domain of our website by encrypting the customers login data into the database servers. Also, we are going to work UI/UX design of

our website by making it catchier and more user-friendly. This are the future tasks we are planning to implement.

8. References

- [1] S. Ma, H. Yang and M. Shi, "Developing a Creative Travel Management System Based on Software Reuse and Abstraction Techniques," 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), 2017, pp. 419-424, doi: 10.1109/COMPSAC.2017.107.
- [2] K. Rutvik Baban, P. Mehra, Sanjay, and J. Manoj, "TOURS AND TRAVEL MANAGEMENT SYSTEM," Peer-Reviewed, Open Access, IRJMETS.
- [3] Onuiri, Ernest & Omoroje, Henry & Ntima, Chukwudi & Omotunde, Ayokunle. (2016). Intelligent Tourism Management System. 18. 304-315.
- [4] Pineda-Jaramillo, Juan. (2019). A review of Machine Learning (ML) algorithms used for modeling travel mode choice. Dyna (Medellin, Colombia). 86. 32-41. 10.15446/dyna.v86n211.79743.
- [5] Aishwarya Bhat, Joylin Rosario, Ayesha Thasneema, Judith Jennifer Rodrigues, 2017, Design and Development of Tour Management System using Android, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCETAIT – 2017 (Volume 5 – Issue 06).