1.1 SUMMARY

Analysis is the process of breaking the problem into the successively manageable parts for individual study, system analysis is the study of various operations that has to be done to solve the problem .one aspect of the system analysis is defining the boundaries of the system and determining whether or not be proposed system should consider other related systems. One of the main meanings of the feasibility is possibility checking of the different criteria for success is included in feasibility study section .these criteria's are cost ,time ,efficiency etc….all these factors play an important role in achieving objective of the system .that means the system should be such it gives optimum performance at minimum cost ,time requirements .these system contributes to the overall objectives of the organization .the system be implemented using current technology and within given cost and schedule constrains .the system is integrated with systems which are already in place. It is a general term that refers to a structural process for identifying and solving problems .in a computer-based transformation system silk is the structured approach. Analysis implies the process of breaching something down in to its parts so that the whole may be understood. The definition of system analysis, but also that of synthesis, which is the process of putting parts together to form a new whole

1.2 PURPOSE

Food is the most important factor that determines human health. Nowadays death and severe health problems are increasing due to poor food safety. With the increase of hotels, the number of food poisoning issues are also increasing and also complaints against hotels are also increasing. For food safety officers Finding genuine complaints and finding hotels with most complaints is difficult. So we developed a web application that will help food safety officers to monitor hotels , that is they can find hotels with most negative reviews and they can also find hotels with positive reviews

The rapid increase in mountains of unstructured textual data accompanied by proliferation of tools to analyse them has opened up great opportunities and challenges for text mining research. The automatic labelling of text data is hard because people often express opinions in complex ways that are sometimes difficult to comprehend. The labelling process involves huge amount of efforts and mislabelled datasets usually lead to incorrect decisions. In this paper, we design a framework for sentiment analysis with opinion mining for the case of hotel customer feedback. Most available datasets of hotel reviews are not labelled which presents a lot of works for researchers as far as text data pre-processing task is concerned. Moreover, sentiment datasets are

often highly domain sensitive and hard to create because sentiments are feelings such as emotions, attitudes and opinions that are commonly rife with idioms, onomatopoeias, homophones, phonemes, alliterations and acronyms. The proposed framework is termed sentiment polarity that automatically prepares a sentiment information extract unbiased opinions of hotel services from reviews.

1.3 TECHNOLOGY

: Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

: Hypertext Mark-up Language (HTML)

It is the standard mark-up language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. HTML can embed programs written in a scripting language such as JavaScript, which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content

: MySQL Server

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used Features of SQL The major features of MySQL Server are:

- Easy installation
- Integration with internet
- Scalability and availabilty
- Support for client/server model
- Operating system capability
- Data replication support
- Full-text search
- Enterprise- level database
- Simplified database administration.

2.1 EXSISTING SYSTEM

Food safety problems are increasing day by day. Nowadays death and severe health problems are increasing due to poor food safety. food and safety officers are unaware about the hygiene and reviews about hotels under their area. There are number of websites to find nearest hotels , hotels in a particular place and to find hotels with maximum ratings and worst ratings, but it is only useful for users who want to choose a hotel. currently there is no websites for food and safety departments to monitor hotels under their area, or to collect complaints or review about hotels. the only way to contact food safety department is only through calls. And most of the time the hygiene is only analysed after a big disaster.

2.2 PROPOSED SYSTEM

Food is the most important factor that determines human health. With the increase of hotels, the number of food poisoning issues are also increasing and also complaints against hotels are also increasing. For food safety officers Finding genuine complaints and finding hotels with most complaints is difficult. So we developed a web application that will help food safety officers to monitor hotels , that is they can find hotels with most negative comments and they can also find hotels with positive reviews and thereby they can monitor hotels under their area and enquire about suspected hotels. There are number of hotels which runs without any proper permissions or certificates .people can find hotels with legal permissions here. They can rate hotels like they do in any other web applications or in google and they can also see the ratings and reviews of hotel.

2.3 FEASIBILITY STUDY

Feasibility is conducted to identify the best system that meets all requirements. It is both necessary and important to evaluate the feasibility of a project at the earliest possible time. feasibility study includes an identification description, an evaluation of proposed system and selection of the best system for the job. During the system is to be carried out. this is to ensure that the proposed system is not A burden to the shop. The feasibility study should be relatively cheap and quick. the results should inform the decision of whether to go ahead with a more detailed analysis, some understanding of the major requirements for the system is essential. Four key considerations involved in the feasibility analysis are

- Technical feasibility
- Economical feasibility
- Behavioural feasibility

- Operational feasibility

2.3.1 TECHNICAL FEASIBILITY

A study of function, performance and constraints may improve the ability to create an acceptable system, technical feasibility is frequently the most difficult area to achieve at the stage of product Engineering process. technical feasibility is deals with the hardware as well as software requirements. the scope was whether the work for the project is done with the current equipment's and the existing system technology has to be examined in the feasibility study. The result was found to be true. This feasibility is carried out to check the technical requirements of the system. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system. this is related to the technicality of the project. This evaluation determines whether the technology needed for proposed system is available or not .it deals with hardware as well as software requirements. that is, type of hardware, software and the methods required for running the system are analyzed .so it can be used in any windows so computer. This system requires very low system resources and it will work in almost all configurations. In the existing system all functions are doing manually. So, if they get this designed software, the problems can be avoided and thus the system will run smoothly.

2.3.2 ECONOMICAL FEASIBILITY

Economic feasibility is the most frequently used method for evaluating the effectiveness of the candidate system .it is very essential because the main goal of the proposed system is to have economically better result along with increased efficiency. A cost evaluation is weighed against the ultimate income or product. Economic justification is generally the bottomline consideration that includes cost benefit analysis, long term corporate income strategies, and cost of resources needed for development and potential market growth. When compared to the advantage obtained from implementing the system its cost is affordable. Proposed system was developed with available resources. Since cost input for the software is almost nil the output of the software is always a profit. Hence software is economically feasible.

2.3.3 BEHAVIOURAL FEASIBILITY

People are inherently resistant to change and computer is known for facilitating the changes .an estimate should be made of how strongly the user; staff reacts towards the development of the computerized system. In the existing system more manpower is required and time factor is more. The

more manpower for managing many files for dynamic data replication and more time for search through these files is needed. but in the proposed system, both manpower and time factors are reduced and also unnecessary burden is reduced. thus, the remaining people are made to engage in some other important work. also, there is no need to wait in case of downloading the data for the users therefore, the system is behaviorally feasible.

2.3.4 OPERATIONAL FEASIBILITY

The purpose of the operational feasibility is to determine whether the new system will be used if it is developed and implemented and whether there will be resistance from users that will undermine the possible application benefits .the aspect of study is to check the level of acceptance of the system by the user .this includes the process of training the user to use the system efficiently .the user must not feel threatened by the system, instead must accept it as a necessity .the level of acceptance by the user solely depends on the methods that are employed to educate the user about the system and to make him familiar with it .his level of confidence must be raised so that he is also able to make some constructive . The proposed system is an upgrade version of the current systems new fields have been implemented according to the user need, hence it ensures all the aspects.

3.1 USER CHARACTERISTICS

consist of only one user

- USER

USER:

Users can register and login. Users can view hotels and they can write their reviews . they can share their experiences with the hotel weather it is good or bad and they can also find hotels with better reviews.

Process done by user:

- Register and login
- View hotels
- Add hotel reviews
- View hotels according to reviews

3.2 SYSTEM CONFIGURATION

3.2.1 HARDWARE CONFIGURATION:

Selection of hardware configuration is very important task related to the software development. The processor should be powerful to handle all the operations. The hard disk should have the sufficient capacity to solve the database and the application.

HARDWARE SPECIFICATION:

CPU	-	Pentium IV Processor And Above
Hard Disk	-	40 GB
Display	-	15”Color Monitor
Main Memory	-	1GB
Keyboard	-	104 Keys
Clock-Speed	-	2.6 GHZ
Monitor	-	15”CRT Monitor

3.2.2 SOFTWARE CONFIGURATION:

SOFTWARE SPECIFICATIONS

Operating System - Windows 10

Web Server - Google Chrome, Internet Explorer, Microsoft Edge

Environment - Python

Front End - HTML

Back End - MySQL SERVER

Browser - Internet Explorer 6 & Above

SYSTEM DESIGN

The term design describes the final system and the way in which it is developed. The system design is a solution, how to approach to the new system. This important phase is composed of several steps. An emphasis is on translating the performance requirements of our proposed system into design specification. Design goes through logical and physical stage of development. In the design phase the physical design producing the working system by defining a particular specification that helps to knowing exactly what the new system must do. The logical design determines the information flow into and of the system and require database. Design is a multistep process that focuses on data structure, software, architecture, procedural details, and interface between modules. The design process translates the requirements into the representation of the software. Computer software design changes continually because new methods, better analysis and broader understanding evolved. It provides the understanding and procedure details necessary for implementing the proposed system .an emphasis is on translating the performance requirement of our proposed system into design specification. Design goes through logical and physical stage. The system design is the last phase that indicate the final system and process of design phase. In the designed phase of maintenance management system, the database tables, input screens and output reports are designed. In table designing, redundancy is avoided. Design is the only way that we can accurately translate a system requirement into a software product. In our production management system, the all-input screens are designed as user friendly and understandable

4.1 INPUT DESIGN

Input design is the link that ties the information system into the world of its users. The input design involves determining what the input is, how the data should be performed, how to validate data, how to minimize data entry and how to provide a multi user facility, inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry operator can be controlled by input design. Input design is the process of converting user originated input to a computer-based format. Input data are collected and organized into groups of similar data. Once identified, appropriate input media are selected for processing

All the input data re validated in the order and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, then it is transferred to the appropriate table in the database. A form is designed to enter the details should be user friendlier so that authorized user with even less knowledge can enter the data. The form is designed using v b tools like command boxes, text boxes, labels, etc.

4.2 OUTPUT DESIGN

Output design is very important concept in the computerized system, without reliable output the user may feel the entire system unnecessary and avoids using it. The proper output design is important in any system and facilitates effective decision making. The output design of this system includes various reports. output requirements are designed during system analysis. An application is successful only when it can provide efficient and effective reports.The goal of the output design is to capture the output and get the data into a format suitable for the computer. It is very helpful to produce the clear, accurate and speedy information for end users.A major form of the output is the harder copy from the pointer and screen reports. Printouts are designed around the output requirements of the user. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. Output of this project is provided in the form of reports created using crystal report tool.

4.3 DATABASE DESIGN

Database is a collection of interrelated data stores with minimum the overall objective in the development of the database technology has been to treat data as an organizational resource and has an integrated whole. Database management system allows data to be protected and organized separately from other resources. Database is an integrated collection of data. this is the difference between logical and physical data. The general objective is to make information access easy, quick, inexpensive and flexible for users. the database approach to system design places greater emphasis on the integration, integrity and independence of data.

4.4 PROCESS DESIGN

Process design represents the structure of data and program components that are required to build a computer-based system. It considered the architectural style that the system will take, the structure and properties of the components constitute the system, and the interrelationships that occur among all architectural components of a system. Although a software engineer candesign both and architecture, the job is often allocated to specialist when large, complex system are to be built. A database or data warehouse designer creates data architecture for a system. The ‘system architect ‘select an appropriate architectural style for the requirements derived during system engineering and software requirement analysis. Architectural design begins with data design and proceeds to the derivation of one or more representations of the architectural structure of the system. An architecture model encompassing data architecture and program structure is created during architectural design. In addition, component properties and the process by which it is developed. It refers to technical specifications that will be applied in implementing the system. It includes the construction of program and program testing. The input to design phase is software requirement specification

4.5 STRUCTURE DESIGN

Structured design deals with the data-flow in the system. It partitions a program into hierarchy of modules. The modules are organized in a top-down manner and the details will be at the bottom. The structured Design begins with a system specification that identifies inputs and outputs that described the functional of the Table.

4.5 DATAFLOW DIAGRAM DESIGN

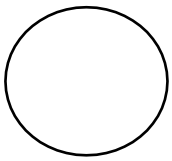
Data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. A DED shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored.

DFD is a designing tool used in the top-down approach to system Design. This context level DFD is next "exploded ", to produce a Level 1 DFD that shows some of the detail of the system being modelled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job and shows the flow of data between the various parts of the system.

- Data Store- collection of data that is permanently stored.
- External Entity- A person, organization or system that is external to the system but interact with it.
- Data Flow- Single piece of data or logical collection of information like a bill.

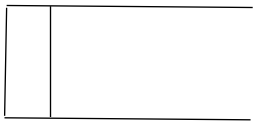
The following are some DFD symbols used in the project.

Rectangle: - It defines a source or destination of system data.



Circle: - It represents a process that transforms incoming data flow into outgoing data flow.

Arrow: - It defines data flow. It is a pipeline through which information flows.



Open rectangle: - It is used to store data or a temporary repository of data

SYSTEM IMPLEMENTATION

A crucial phase in the system life cycle is the successful implementation of the new system design. Implementation involves creating computer compatible files, training the operating staff, installing hardware, terminals. In the system implementation, user training is crucial for minimizing resistance to change and giving the new system a chance to prove its worth. The objectives of the system implementation are to put the system into operation while holding costs, risks and personal irritation to minimum. Once the physical system has been designed in details, the next stage is to run the design into a working system and then to monitor the operation of the system to ensure that is continue to work efficiently and the operation of the system to ensure that is continues to work efficiently and effectively. The implementation stage of is often very complex and time consuming because many more people are involved than in the earlier stages. The system implementation took place through various stages as follows,

- Implantation planning.
- Education and training.
- System testing.
- System implementation.
- Change over.

The implementation plan includes a description of all the activities that must occurs to implement the new system and to put it into operation. To achieve the objectives and benefits from computerbased system, it is essential for the people who will be confident of their role in the new jobs. After software is developed to meet user's requirements, users test it for acceptance. The changes over phase are used to provide adaptability for the new system.

5.1 MODULE SPECIFICATION

consist of two modules, and they are

- ADMIN
- USER

ADMIN:

Admin is the food and safety officer. Each and every process that carried out in this system are under the supervision of administrator. Admin adds hotels and images of kitcehens and hotels. Admin can also view the reviews and the final reviews

Process done by the administrator:

- Add hotels
- Upload images
- view images
- View hotels
- View reviews of users
- View hotels after

USER:

Users can view hotels and they can write their reviews . they can share their experiences with the hotel weather it is good or bad and they can also find hotels with better reviews.

Process done by user:

- Register and login
- View hotels
- Add hotel reviews
- View hotels according to reviews

5.2 SAMPLE CODING

```
from django.core.files.storage import FileSystemStorage

from django.db import connection

from django.http import HttpResponse

from django.shortcuts import render, redirect

from textblob import TextBlob

from datetime import date

from django.contrib.auth import logout


# Create your views here.


def index(request):

return render(request,&#39;user/index.html&#39;)


def adminindex(request):

return render(request,&#39;admindashboard/index.html&#39;)


def login(request):

return render(request,&#39;login.html&#39;)


def login1(request):

if request.method == &#39;POST&#39;:

userid = request.POST[&#39;username&#39;]

password = request.POST[&#39;password&#39;]


cursor = connection.cursor()

cursor.execute(&quot;select * from login where admin_id=&#39;&quot; + userid + &quot;&#39;
and password=&#39;&quot; + password

+ &quot;&#39;&quot;)


admin = cursor.fetchone()

if admin == None:
```

cursor.execute("select * from user_register where name='" + userid + "' and password='" +

password + "'")

user = cursor.fetchone()

if user == None:

return redirect("login")

else:

request.session['userid'] = userid

return render(request, 'user/UserHeaderPage.html')

else:

return render(request, 'admindashboard/index.html')

def userfooterpage(request):

return render(request,'user/UserFooterPage.html')

def userheaderpage(request):

return render(request,'user/UserHeaderPage.html')

def hotelregistration(request):

if request.method == 'POST':

hotelid =request.POST['name']

name = request.POST['name']

address = request.POST['address']

phone =request.POST['phone']

email =request.POST['email']

districtid = request.POST['districtid']

taluk = request.POST['taluk']

regno = request.POST['regno']

cursor = connection.cursor()

cursor.execute("select * from hotel_registration where hotel_id = '" + str(hotelid) + "' ")

hotel = cursor.fetchone()

if hotel == None:

cursor = connection.cursor()

cursor.execute("insert into hotel_registration values('"+ str(hotelid) + "','"+ str(name) + "','"+

str(address) + "','"+ str(phone) + "','"+ str(email) + "','"+ str(districtid) + "','"+ str(taluk) +

"','pending','"+ str(regno) + "')")

return redirect('userheader')

else:

return HttpResponse("<script>alert('User Name already

exists');window.location='../hotelregistration';</script>")

else:

return render(request,'user/hoter_registration.html')

def rating(request,id):

request.session['hotid'] = id

print(id)

return render(request,'user/rating.html')

def rate(request):

if request.method == 'POST':

print("_____")

hid = request.session['hotid']

userid = request.session['userid']

rating = request.POST['rating']

cursor = connection.cursor()

cursor.execute("insert into rating values(null,'" + str(hid) + "','" + str(userid) + "','" + str(rating) +

"')")

```
return redirect(&#39;userheader&#39;)

def review(request,id):
request.session[&#39;hotid&#39;] = id
return render(request, &#39;user/review.html&#39;)

def addkitchenimages(request):
if request.method == &#39;POST&#39;:
name = request.POST[&#39;userid&#39;]
hotelid = request.POST[&#39;hotelid&#39;]
date = request.POST[&#39;date&#39;]
image = request.FILES[&#39;image&#39;]
fss = FileSystemStorage()
file = fss.save(image.name, image)
file_url = fss.url(file)
description = request.POST[&#39;description&#39;]
cursor = connection.cursor()

cursor.execute(&quot;insert into kitchen_images
values(null,&#39;&quot;+str(name)+&quot;&#39;,&#39;&quot;+str(hotelid)+&quot;&#39;,&#39;
&quot;+str(date)+&quot;&#39;,&#39;&quot;+str(image)+&quot;&#39;,&#39;&quot;+str(descript
ion)+&quot;&#39;,&#39;pe
nding&#39;)&quot;)
return redirect(&#39;addkitchenimages&#39;)
else:
return render(request,&#39;admindashboard/add_kitchen_images.html&#39;)

def viewkitchen(request):
cursor = connection.cursor()
cursor.execute(&quot;select * from kitchen_images&quot;)
all=cursor.fetchall()
return render(request,&#39;user/viewkitchen.html&#39;,{&#39;data&#39;:all})
```

```
def userheader(request):
return render(request,&#39;user/UserHeaderPage.html&#39;)

def userfooter(request):
return render(request,&#39;user/UserFooterPage.html&#39;)

def userregister(request):
if request.method == &#39;POST&#39;:
userid = request.POST[&#39;name&#39;]
name = request.POST[&#39;name&#39;]
address = request.POST[&#39;address&#39;]
email = request.POST[&#39;email&#39;]
phone = request.POST[&#39;phone&#39;]
password = request.POST[&#39;password&#39;]

cursor = connection.cursor()
cursor.execute(&quot;select * from user_register where user_id = &#39;&quot; + str(userid) + &quot;&#39; &quot;)
user = cursor.fetchone()

if user == None:
cursor = connection.cursor()
cursor.execute(&quot;insert into user_register
values(&#39;&quot;+str(userid)+&quot;&#39;,&#39;&quot;+str(name)+&quot;&#39;,&#39;&quot;+str(address)+&quot;&#39;,&#39;&quot;+str(email)+&quot;&#39;,&#39;&quot;+str(phone)+&quot;&#39;,&#39;&quot;+str(pass
word)+&quot;&#39;)&quot;)
request.session[&#39;userid&#39;]=userid
return redirect(&#39;userheader&#39;)
else:
return HttpResponse(&quot;&lt;script&gt;alert(&#39;User Name already
exists&#39;);window.location=&#39;../register&#39;;&lt;/script&gt;&quot;)

else:
```

```
return render(request,&#39;user/registerartion.html&#39;)

def viewreview(request):
cursor = connection.cursor()
cursor.execute(&quot;select * from review&quot;)
all = cursor.fetchall()
return render(request,&#39;admindashboard/viewreview.html&#39;,{&#39;data&#39;:all})

def viewrating(request):
cursor = connection.cursor()
cursor.execute(&quot;select * from rating&quot;)
all = cursor.fetchall()

return render(request,&#39;admindashboard/viewrating.html&#39;,{&#39;data&#39;:all})

def deleterating(request,id):
cursor = connection.cursor()
cursor.execute(&quot;delete from rating where idrating =&#39;&quot;+str(id)+&quot;&#39;&quot;)
return redirect(&#39;viewrating&#39;)

def deletereview(request,id):
cursor = connection.cursor()
cursor.execute(&quot;delete from review where idreview =&#39;&quot;+str(id)+&quot;&#39;&quot;)
return redirect(&#39;viewreview&#39;)

def review1(request):
if request.method == &#39;POST&#39;:
hotel_id = request.session[&#39;hotid&#39;]
review_details = request.POST[&#39;review_details&#39;]
cursor = connection.cursor()
cursor.execute(&quot;insert into review values(null,&#39;&quot; + str(review_details) + &quot;&#39;,curdate(),&#39;&quot; +
```

```
str(hotel_id) + &quot;&#39;)&quot;)

feedback = review_details

# print text

print(feedback)

obj = TextBlob(feedback)

# returns the sentiment of text

# by returning a value between -1.0 and 1.0

sentiment = obj.sentiment.polarity


print(sentiment)

if sentiment == 0:

print(&#39;The text is neutral&#39;)

cursor = connection.cursor()

cursor.execute(&quot;select * from review_nltk where hotel_id=&#39;&quot; + hotel_id + &quot;&#39; &quot;)

pins = cursor.fetchone()

if pins == None:

cursor = connection.cursor()

cursor.execute(&quot;insert into review_nltk values(null,0,0,1,&#39;&quot; + hotel_id + &quot;&#39;)&quot;)

else:

cursor = connection.cursor()

cursor.execute(

&quot;update review_nltk set neutral=neutral+1 where hotel_id=&#39;&quot; + hotel_id + &quot;&#39; &quot;)

elif sentiment &gt; 0:

print(&#39;The text is positive&#39;)

cursor = connection.cursor()

cursor.execute(&quot;select * from review_nltk where hotel_id=&#39;&quot; + hotel_id + &quot;&#39; &quot;)

pins = cursor.fetchone()

if pins == None:

cursor = connection.cursor()
```

cursor.execute("insert into review_nltk values(null,1,0,0,'" + hotel_id + "')")

else:

cursor = connection.cursor()

cursor.execute(

"update review_nltk set positive_review=positive_review+1 where hotel_id='" + hotel_id

+ "' ")

else:

print('The text is negative')

cursor = connection.cursor()

cursor.execute("select * from review_nltk where hotel_id='" + hotel_id + "' ")

pins = cursor.fetchone()

if pins == None:

cursor = connection.cursor()

cursor.execute("insert into review_nltk values(null,0,1,0,'" + hotel_id + "')")

else:

cursor = connection.cursor()

cursor.execute(

"update review_nltk set negative_review=negative_review+1 where hotel_id='" +

hotel_id + "' ")

return redirect('userheader')

else:

return redirect('review')

def logout1(request):

logout(request)

return redirect('../')

def viewhotelregistration(request):

```
cursor = connection.cursor()

cursor.execute(&quot;select * from hotel_registration where status = &#39;pending&#39;&quot;)

hotels = cursor.fetchall()

return
render(request,&#39;admindashboard/view_hotel_registration.html&#39;,{&#39;data&#39;:hot
els})


def approve(request,hotelid):

cursor = connection.cursor()

cursor.execute(&quot;update hotel_registration set status = &#39;approve&#39; where hotel_id
=

&#39;&quot;+str(hotelid)+&quot;&#39; &quot;)

return redirect(&#39;viewhotelregistration&#39;)


def approvedhotel(request):

cursor = connection.cursor()

cursor.execute(&quot;select * from hotel_registration where status = &#39;approve&#39;
&quot;)

hotels = cursor.fetchall()

return
render(request,&#39;admindashboard/view_approved_hotel.html&#39;,{&#39;data&#39;:hotel
s})


def viewnltkreview(request):

cursor = connection.cursor()

cursor.execute(&#39;select * from review_nltk&#39;)

all = cursor.fetchall()

return
render(request,&#39;admindashboard/view_review_nltk.html&#39;,{&#39;data&#39;:all})


def deletehotel(request,id):

cursor = connection.cursor()

cursor.execute(&quot;delete from hotel_registration where hotel_id =
&#39;&quot;+str(id)+&quot;&#39;&quot;)

return redirect(&#39;viewapprovedhotel&#39;)
```

SYSTEM TESTING

Testing is an important phase in software development. After completion, the system may work without any problem. But there should be several unknown or hidden errors in the system still remaining. The error chances may be injected into the system at any stage of the development. Even if there are techniques to detect and eliminate the errors, some errors may retain in the system. So, after the completion of coding, the system is to be executed with the only purpose of detecting maximum number of errors. The tester executes the system, and inputs different types of values those may cause error or some exceptional situation in the system. The error locations detected through the testing are to be corrected in the system then. So, the important and the only aim of testing is to detect and cure even a less possible of an error that may face in the future executions of the system. Testing is a set of activity that can be planned in advance and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is completed without testing, as it is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are corrected, the goal will be successfully achieved. Inadequate testing or non-testing may lead to errors that may not appear until months later.

6.1 PURPOSE OF TESTING

Testing is the success of the system. System testing makes a logical assumption that if all part of the system is correct, the goal will be successfully achieved. The following points shows how testing is essential. Existence of program defects of inadequacies is inferred. Verifies whether the software behave as intended by its designer. Checks conformance with requirements specification or user need.

Access the operational reliability of the system. Test the performance of the system. The performance of the system. Reflects the frequencies of actual user inputs. Find the fault which caused the output anomaly. Detect flaws and deficiencies in requirements. Exercise the program using data like the real data processed by the program. Test the system capabilities. Judges whether or not the program is usable in practice. Testing objectives there are several rules that can serve as testing objectives. They are; Testing is a process of executing a program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. A successful test is one that uncovers an undiscovered error.If testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrates that software functions appear to the working according to the specifications, that performance requirement appear to have been met. These are three ways to test a program for correctness for implementation efficiency for computational complexity Test for correctness are supported to verify that a program does exactly what it was designed to do. This is much difficult that it may at first appear especially for large programs.Tests for implementation efficiency attempt to find ways to make a correct program faster or use less storage. It is a code-refining process, which reexamines the implementation phase algorithm

development. Tests for computational complexity amount to an experiment analysis of the complexity of an algorithm or an experiment comparison of two or more algorithms, which solve the same problem.

6.2 TYPES OF TESTING

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct. The goal will be successfully achieving. The candidate system is subject to a variety of tests. A series of tests are performed for the proposed system is ready for system acceptance testing. The various levels at which testing are conducted are,

- Unit testing
- Integration testing
- Sequential testing
- System testing
- Validation testing unit testing

6.2.1 UNIT TESTING

In unit testing each program unit is tested individually.so any errors in a unit are debugged. Sample data is given for unit testing. The unit test results are recorded for future references. Unit testing focus verification efforts on the smallest unit of software design, the module. This is known as "module testing". It comprises of the set test performed by an individual programmer prior to the integration of unit into the large system. The modules are tested separately, this testing is carried out programming stage itself.In this step each module is found to be working satisfactory as regard to the expected out from module. The unit testing was done for every module in the software for various inputs, such they each line of code is at least once executed. This testing was carried out during the unit to a large system.

6.2.2 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing.

6.2.3 PROGRAM TESTING

Program testing checks for two types of errors; syntax and logic. A syntax error is a program statement that violates one or more rules of the language in which it is written. A logic error deals with incorrect data fields. When a program is tested, the actual output is compared with the expected output. All the modules are combined and tested as a whole. Here correction is difficult because the vast expenses of all errors uncovered are correct for the next testing steps. We follow bottom-up integration. Bottom-up integration testing as its name implies begin construction and sling with atomic modules. Because components are integrated from the bottom up, accessing required for the components subordinate to a given level is always available and need for stubs is eliminated.

6.2.4 SEQUENTIAL TESTING

Sequential or series testing is checking the logic of one or more programs in the candidate system, where the output of one program will affect the processing done by another program.

6.2.5 SYSTEM TESTING

System testing executing a program to check logic changes made in it and with the intension of finding errors-making the program fails. Effective testing does not guaranty reliability is a design consideration. This testing actually consists of a series of different test whose primary purpose is to fully exercise the computer based system.it begins where integration testing is completed and finally software is completely assembled as package, interfacing errors are uncovered and corrected.

6.2.6 ACCEPTANCE TESTING

Acceptance testing is running the system with live data by the actual user. An acceptance test has the objective of selling the user in the validity and reliability of the system. A comprehensive test report is prepared. The report indicates the system's tolerance, performance range, error rate and accuracy. It verifies the system procedures operate to system specification and the integrity of important data is maintained, performance of an acceptance test is actually the users show. User motivation is very important for the successful performance of the system. After that a comprehensive test report is prepared. This report shows the systems tolerance, performance range, error rate and accuracy.

6.2.7 INPUT TESTING

Here system is tested with all verifiable combination of input. User may type data in situations like entering password, numerical details etc. The system is tested with all the causes and it responded with appropriate error message.

6.2.8 OUTPUT TESTING

Here the output is tested to view where the screen is what which is desired. It is also checked whether it is to the satisfaction of the user. Changes that need to be done can be done after the result is seen

SYSTEM MAINTENANCE

Software maintenance is the process of modifying a software system or component after its delivery in order to correct faults improve the performance and other attributes, or to adapt to the changed environment. maintenance covers a wide range of activities including correcting the error and design coding, updating the documentation and test data, and upgrading the user hardware and software. maintenance is always necessary to keep the software usable and useful. Hardware also requires periodic maintenance to keep the system into its standards. After installation is completed and user start is adjusted to the changes created by the candidate system. evaluation and maintenance begin. If new information is consistent with design specification the changes have to be made. Hardware also requires periodic maintenance to keep in tune with design specifications. User priorities changes in organizational requirements or environmental factors also called for system enhancements. Maintenance covers wide range of activities, including correcting, coding and design errors, updating documentation and test data, and upgrading user support. any activities classified as maintenance are actually enhancements. Maintenance means restoring something to do its original condition. Unlike hardware, software does not wear out; it is corrected.in contrast, enhancement means adding, modifying or redeveloping the code to support changes in the specifications.it is necessary to keep up with changing user needs the operational environment. Maintenance means repairing processing or performance failures or making changes because of previously uncorrected problems or false assumptions. adaptive maintenance means changing the program function. Perfective maintenance means enhancing the performance or modifying the program to respond to the user's additional or changing needs. Of these types, more time and money are spent on perfective than on corrective and adaptive maintenance together. Maintenance activities begin where conversion leaves off. Maintenance is handled by the same planning and control used in a formal system project. a major problem with Software maintenance is its labor-intensive nature. Documentation is as much a part of maintenance as it is of system development .to put maintenance in its proper perspective requires considerable skill and experience and is an important and is an important and ongoing aspect of system development .an additional factor in the success of the maintenance programmer is the work environment.

Maintenance Types

System maintenance can be classified into four types –

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance
- Preventive Maintenance

7.1 CORRECTIVE MAINTENANCE

Corrective Maintenance deals with the repair of faults or defects found in day- today system functions. A defect can result due to errors in software design, logic and coding. Design errors occur when changes made to the software are incorrect, incomplete, wrongly communicated, or the change request is misunderstood. Logical errors result from invalid tests and conclusions, incorrect implementation of design specifications, faulty logic flow, or incomplete implementation of design specifications, faulty logic flow, or incomplete test of data. All these errors, referred to as residual errors, prevent the software from confirming to its agreed specifications. Note that the need for corrective maintenance is usually initiated by big reports drawn by the users.

7.2 ADAPTIVE MAINTENANCE

Adaptive Maintenance is the implementation of changes in a part of the system, which has been affected by a change that occurred in some other part of the system. Adaptive Maintenance consists of adapting software to changes in the environment such as the hardware or the operating system. The term environment in this context refers to the conditions and the influences which act (from outside) on the system. For example, business rules, work patterns and government policies have a significant impact on the software system.

7.3 PERFECTIVE MAINTENANCE

Perfective Maintenance mainly deals with implementing new or changed user requirements. Perfective Maintenance involves making functional enhancements to the system in addition to the activities to increase the system's performance even when the changes have not been suggested by faults. This includes enhancing both the function and efficiency of the code and changing the functionalities of the system as per the users' changing needs.

7.4 PREVENTIVE MAINTENANCE

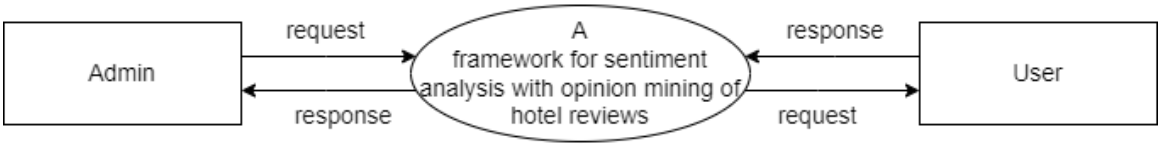
Preventive Maintenance involves performing activities to prevent the occurrence of errors. It tends to reduce the software complexity thereby improving program understand ability and increasing software maintainability. It comprises documentation updating, code optimization and code restructuring. Documentation updating involves modifying the documents affected by the changes in order to correspond to the present state of the system. Code optimization involves modifying the programs for faster execution or efficient use of storage space. Code restructuring involves transforming the program structure for reducing the complexity in source code and making it easier to understand.

CONCLUSION

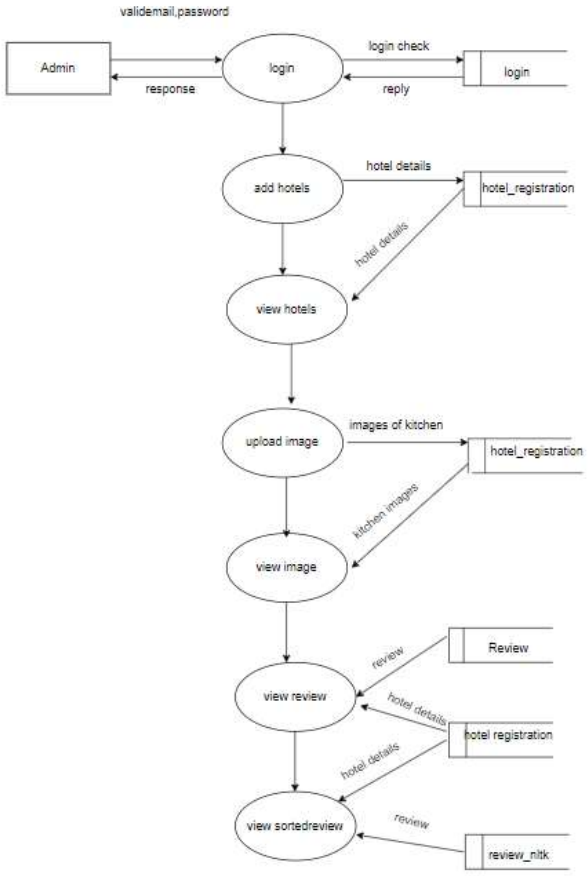
Food is the most important factor that determines human health. Nowadays death and severe health problems are increasing due to poor food safety. food and safety officers are unaware about the hygiene and reviews about hotels under their area. So we developed a web application that will help food safety officers to monitor hotels , that is they can find hotels with most negative comments and they can also find hotels with positive reviews and thereby they can monitor hotels under their area and enquire about suspected hotels. There are number of hotels which runs without any proper permissions or certificates .people can find hotels with legal permissions here. They can rate hotels like they do in any other web applications or in google and they can also see the ratings and reviews of hotel.

9.1 DATA FLOW DIAGRAMS:

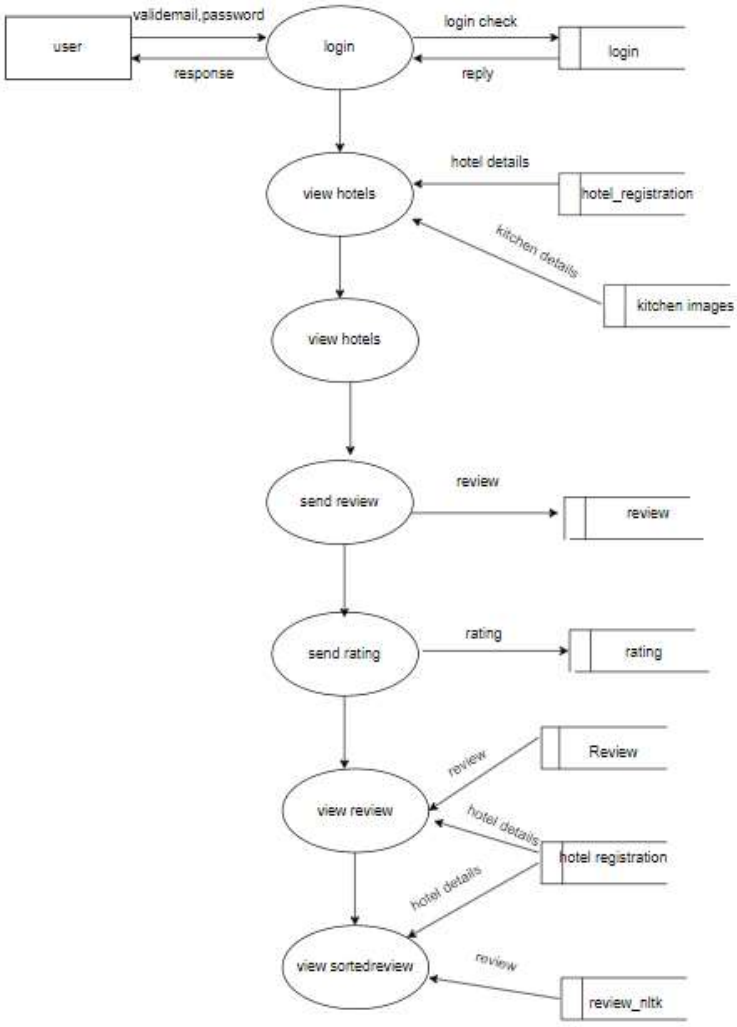
LEVEL- 0 DFD:



LEVEL- 1 ADMIN DFD:



LEVEL- 1 USER DFD:



9.2 DATABASE TABLES:

hotel_registration

FIELD NAME	TYPE	CONSTRAINS	DESCRIPTION
hotel_id	Varchar(45)	Primary key	Unique id to identify hotel
name	Varchar(45)	Null	Name of hotel
address	Varchar(45)	Null	Address of hotel
phone	Varchar(45)	Null	Phone number of hotel
email	Varchar(45)	Null	Email of hotel
district_id	Varchar(45)	Null	District id of hotel
taluk_id	Varchar(45)	Null	Taluk id of hotel
stastus	Varchar(45)	Null	Status of hotel
registration_no	Varchar(45)	Null	Registration number of hotel

Kitchen_images

FIELD NAME	TYPE	CONSTRAINS	DESCRIPTION
Idlitchen_images	Int	Primary key	Unique key for kitchen
user_id	Varchar(45)	Null	Users id
hotel_id	Varchar(45)	Null	Hotel id
upload_date	Varchar(45)	Null	Upload date
file_path	Varchar(45)	Null	File_path
file_description	Varchar(45)	Null	File_description
status	Varchar(45)	Null	Status

login

FIELD NAME	TYPE	CONSTRAINS	DESCRIPTION
admin_id	Varchar(45)	Primary key	Id
Password	Varchar(45)	Null	password

rating

FIELD NAME	TYPE	CONSTRAINS	DESCRIPTION
idrating	Int(10)	Primary key	Unique key
hotel_id	Varchar(45)	Null	Hotel id
user_id	Varchar(45)	Null	User id
rating	Varchar(45)	Null	rating

review

FIELD NAME	TYPE	CONSTRAINS	DESCRIPTION
Idreview	Int(10)	Primary key	Unique key
review_detaoils	Varchar(45)	Null	review detail
review_date	Varchar(45)	Null	reviewdate
hotelid	Varchar(45)	Null	hotelid

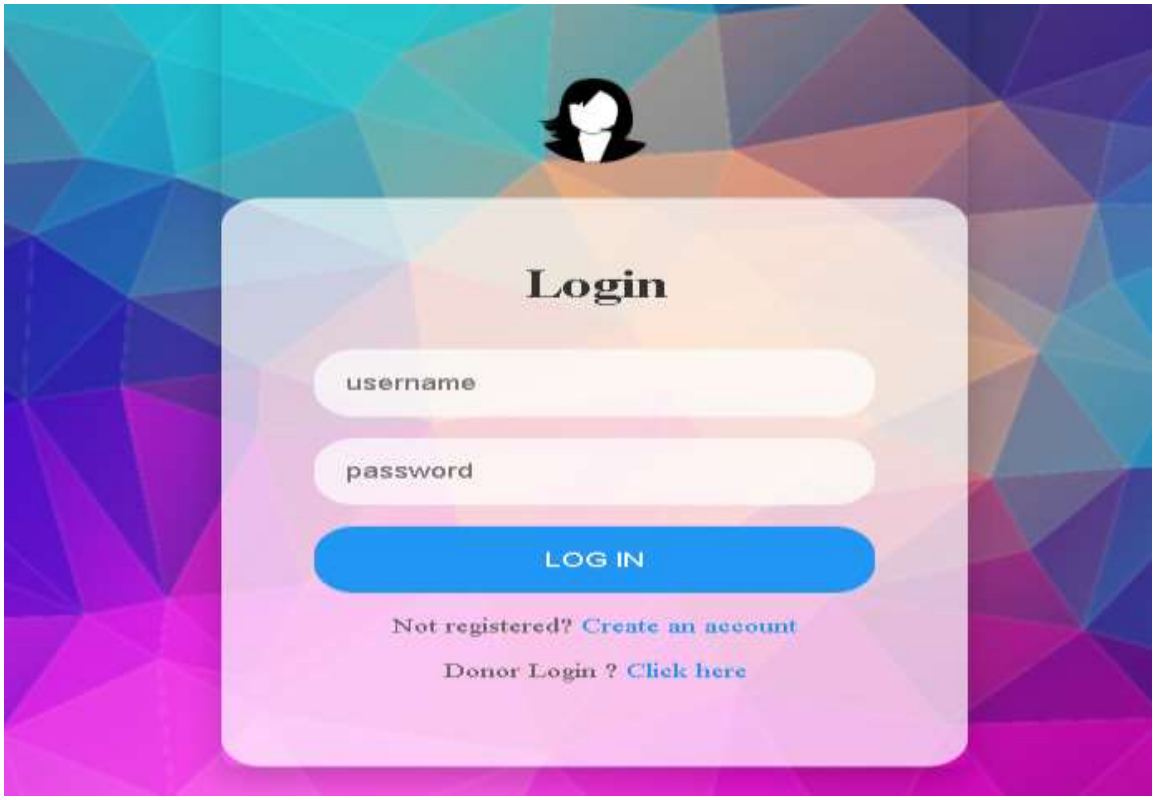
review_nltk

FIELD NAME	TYPE	CONSTRAINS	DESCRIPTION
Idreview_nltk	Int(10)	Primary key	Unique key
Positive_reviewls	Varchar(45)	Null	Positivereview
negative_review	Varchar(45)	Null	Negative reviews
neutral	Varchar(45)	Null	Neutral reviews
hotelid	Varchar(45)	Null	hotelid

User_register

FIELD NAME	TYPE	CONSTRAINS	DESCRIPTION
user_id	Varchar(45)	Primary key	Unique id to identify user
name	Varchar(45)	Null	Name of user
address	Varchar(45)	Null	Address of user
phone	Varchar(45)	Null	Phone number of user
email	Varchar(45)	Null	Email of user
password	Varchar(45)	Null	password

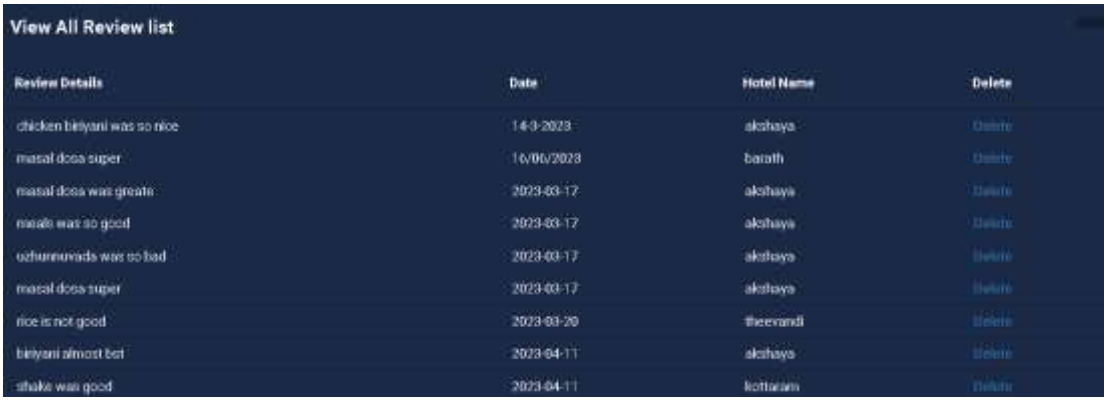
9.3 SAMPLE INPUT PAGES:




Hotel
Akshaya
akshaya

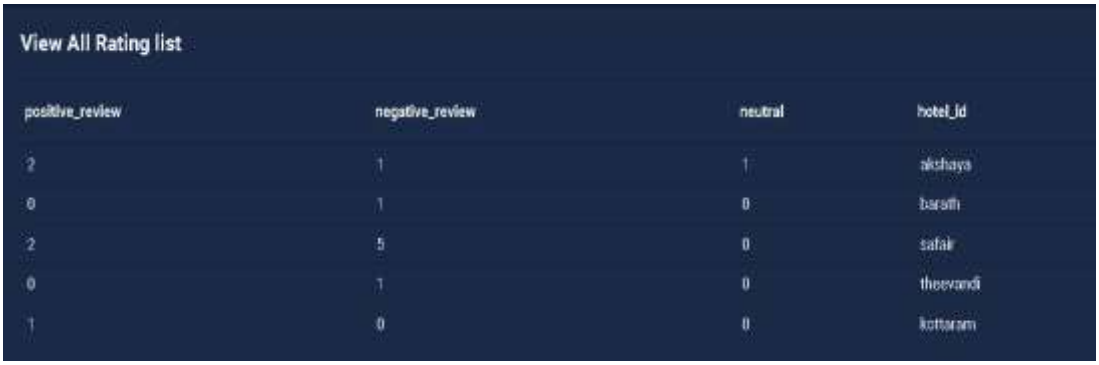

Rating
Rating
submit


Hotel Registration
Approved Hotel
Add kitchen images
View AI Review
View AI Rating
NLTK Review
View All Rating list
Hotel Id
Name
address
phone
email
district_id
taluk_id
registration_no
status



View All Review list

Review Details	Date	Hotel Name	Delete
chicken biriyani was so nice	14-3-2023	akshaya	Delete
masal dosa super	16/06/2023	barath	Delete
masal dosa was greate	2023-03-17	akshaya	Delete
masala was so good	2023-03-17	akshaya	Delete
uzhunnuvada was so bad	2023-03-17	akshaya	Delete
masal dosa super	2023-03-17	akshaya	Delete
rice is not good	2023-03-20	theevandi	Delete
biriyani almost bst	2023-04-11	akshaya	Delete
shake was good	2023-04-11	kottaram	Delete



View All Rating list

positive_review	negative_review	neutral	hotel_id
2	1	1	akshaya
0	1	0	barath
2	5	0	safair
0	1	0	theevandi
1	0	0	kottaram

REFERENCES

The following materials were refered to develop the project Predict Medical Insurance :-

1.Software Engineering A Practitioner's Approach 7th Edition

By Roger.Pressman

2.Database System Concepts 7th Edition

By Abraham Silberschatz,

Henry F.Korth

S.Sudarshan

3.Data Mining Concepts and Techniques 3rd Edition

By Jiawei Han

Micheline Kamber

Jian pei

4.https://www.javatpoint.com

5.http://www.admecindia.co.in

6. https://www.telusko.com