

MID SEM REVIEW

Presentation On SEMESTER LONG INTERNSHIP DESIGN

Guided by, Internal Guide – Mr. B. R. Patil Industry Mentor – Mr. Ajit Sutar (Lead Developer)

Presented By-Prajwal Pradeep Agarwal-4203 Role- Backend Developer

ABOUT COMPANY

COMPSERV CONSULTANCY PRIVATE LIMITED

CCPL is a Decision Support System Provider to Business Leaders. They implement innovative and sustaining IT solutions for their Clients which enables them to save skilled man-hours and achieve optimum utilization of their resources using most efficient technologies, methods and processes through their ERP solution BWays.

You will get Compserv's derived and proven solution for Decision Support System for Business Leader based on their 150+ man years' experience as our core team is implementing and developing these solutions for various industries from last 28 years.







ROADMAP TIL DATE



Hand-On experience working on technology

Training Period

- 1. Python
- 2. MySQL
- 3. FastAPI

Selection of Problem Statement and proposed Solution

Creation of Flow towards design of project





TRAINING PERIOD



Python

- Code Readability
- Large Standard Library
- Object-oriented **Programing**
- portability





3rd week of Jan

MySQL

- Scalable and manages memory
- Fast and easy to use
- Highly used for data analysis



4th week of Jan 1st week of Feb

FastAPI

- In-build Data Validation
- **Automatic Documentation** Support -swagger UI

CERTIFICATES AND ACHIVEMENTS



This is to certify that Prajwal Pradeep Agarwal successfully completed and received a passing grade in Python (CEPYTIIN, provided by IBMCE) A course on etrain.skillsnetwork.site Powered by IBM Developer Skills Network. Issued by March 2, 2023 Authenticity of this certificate can be validated by going to:

- Advanced level certificate of SQL on Hacker-rank
- Golden Badge for SQL on Hacker-Rank Platform





- IBM Skill platform certificate for python
- Golden Badge for SQL on Hacker-Rank Platform

PROJECT ON CHATBOT



- Existing Support System Problem solving team through telephonic Conversation and large number of pending calls.
- Proposed Solution Chatbot System with 40% problem solving using knowledge base and 60% through Admin panel.

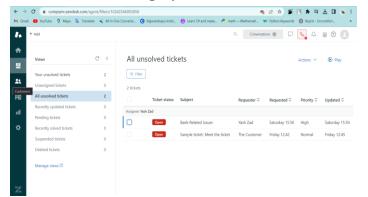
CHATBOT WITH KNOWLEDEGE BASE-

- Problem Solving using database
- Fast and Accurate Solution
- High Efficiency in numbers
- Automation in conversation
- Training of bot using machine Learning

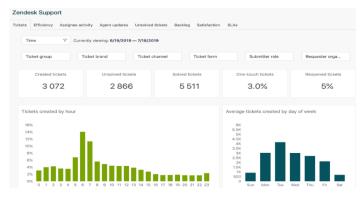


LITERATURE OF EXISTING PLATFORM

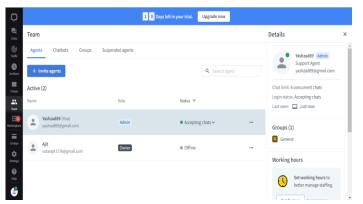
ZenDesk Ticketing System



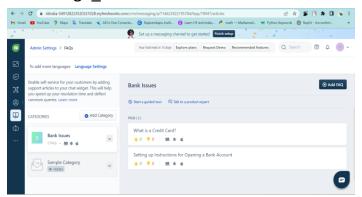
ZenDesk Report and Analytics



Agent adding & dashboard option in LiveChat



Knowledge base in Freshworks

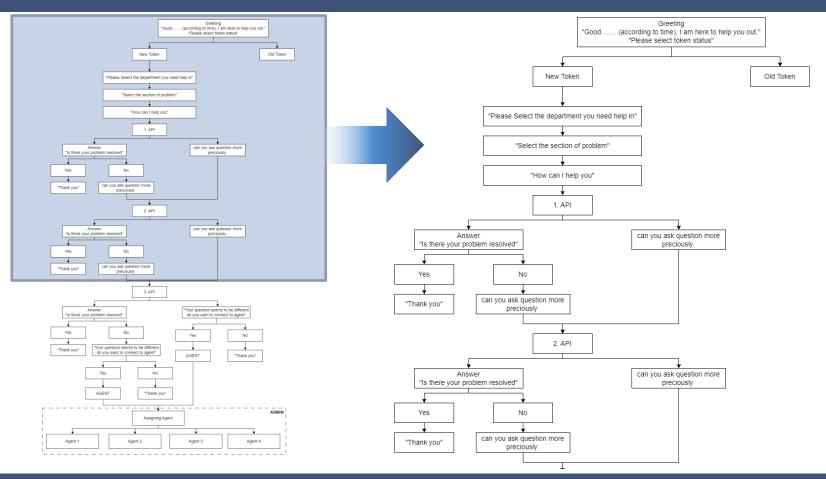




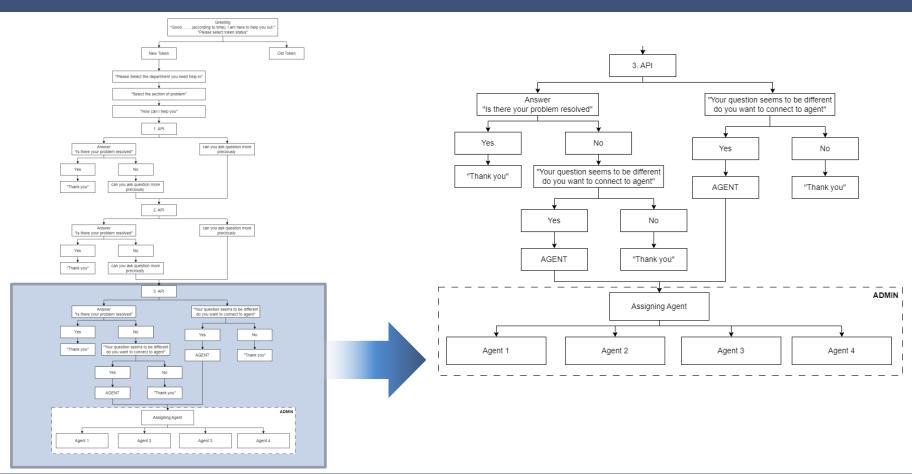




PROCESS FLOW OF PROJECT



PROCESS FLOW OF PROJECT



SNAPSHOT – TEXT MATCH SEARCH ALGO

```
import mysql.connector
import nltk
cnx = mysql.connector.connect(
  host="localhost",
  user="root",
  port =3306,
  database="chatbot",
  password = 'soulreaper3000'
cursor = cnx.cursor(buffered=True)
query = "SELECT * FROM knowledge bases where department = '{a}' and section = '{b}'".\
  format(a = input("In which Deaprtment you have problem:\n"),
         b = input(" Please let me know the section:\n"))
query1 = input("Please enter the query string:\n")
cursor.execute(query)
rows = cursor.fetchall()
cursor.close()
cnx.close()
linklist = []
for i in rows:
  list = []
  list.append(i[3])
  list.append(i[4])
```

```
linklist = []
for i in rows:
 list = []
 list.append(i[3])
 list.append(i[4])
 def mathingWords(sentence1, sentence2):
   words1 = sentence1.strip().split(" ")
   words2 = sentence2.strip().split(" ")
   score = 0
   for word1 in words1:
       for word2 in words2:
           if word1.lower() == word2.lower():
               score += 1
   return score
 if name == " main ":
     sentences = list
     #print(sentences)
     scores = [mathingWords(query1, sentence) for sentence in sentences]
     sortedSentScore = [sentScore for sentScore in sorted(
          zip(scores, sentences), reverse=True) if sentScore[0] !=0 ]
     for score, item in sortedSentScore:
         if score >= 1:
           linklist.append(i[5])
print(linklist[0:3], sep=", ")
```

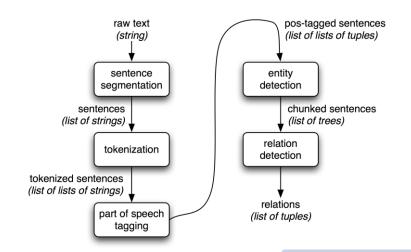
NLTK- NATURAL LANGUAGE TOOLKIT

Snapshot- Sample code of NLTK

```
import numpy as np
import nltk
#nltk.download('punkt')
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()
def tokenize(sentence):
    return nltk.word tokenize(sentence)
def stem(word):
    return stemmer.stem(word.lower())
def bag of words(tokenized sentence, words):
    sentence words = [stem(word) for word in tokenized sentence]
    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
        if w in sentence words:
            bag[idx] = 1
    return bag
```

NLTK – Natural Language Toolkit-

- NLTK is a leading platform for building Python programs to work with human language data.
- along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries



WEBSOCKETS

WebSocket is a communication protocol that provides full-duplex, real-time communication between a client and a server over a single TCP connection. It enables efficient and low-latency data exchange, making it suitable for applications that require real-time updates, such as chat applications, real-time collaboration tools, and streaming applications.

Handshake (HTTP upgrade)

connection opened

Bi-directional messages

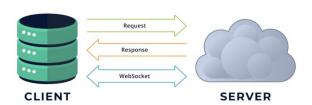
open and persistent connection

One side closes channel

connection closed

- Integration
- Endpoint Definition
- Bi-directional Communication
- Asynchronous Support
- Accepting Connections
- Data Exchange

WebSocket Connection





THANK YOU