AKENTEN APPIAH-MENKA UNIVERSITY OF SKILLS TRAINING AND ENTREPRENEURIAL DEVELOPMENT

IMPLEMENTATION OF SPEECH RECOGNITION IN POINT-OF-SALE SYSTEM

STEPHEN KWAKU POMETSEY

2023

AKENTEN APPIAH-MENKA UNIVERSITY OF SKILLS TRAINING AND ENTREPRENEURIAL DEVELOPMENT

IMPLEMENTATION OF SPEECH RECOGNITION IN POINT-OF-SALE SYSTEM

STEPHEN KWAKU POMETSEY

(7201040003)

A dissertation in the department of Information Technology Education, Faculty of Technical Education submitted to the school of graduate studies in partial fulfilment

of the requirements for the award of the degree of

Master of Science

In Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development

2023

..

STUDENT'S DECLARATION

I, Stephen Kwaku Pometsey declare that this thesis, with the exception of quotations and references contained in published works which have all been identified and duly acknowledged, is entirely my own original work, and it has not been submitted, either in part or whole, for another degree elsewhere.

SIGNATURE…….

DATE…….

SUPERVISOR'S DECLARATION

I / We hereby declare that the preparation and presentation of this work was supervised in accordance with the guidelines for supervision of thesis/dissertation/project as laid down by the Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development.

…….. (Principal Supervisor)

Signature ……

Date……

…… (Co-Supervisor)

Signature ……

Date……

DEDICATION

I dedicate this research study to the Almighty God for the protection, love and the knowledge he has given me. Nothing could have been done without him.

Also, I dedicate this to my parents Mr. Charles Pometsey and Mrs. Agnes Pometsey for their strict and lovely parenting which has really brought me this far. The project is also dedicated to my brothers and my lovely wife Mrs. Margaret Pometsey and my lovely children, Elorm Yvonne Pometsey, Eyram Anissa Pometsey and Edudzi Jace Junior Kwaku Pometsey.

May God bless you all.

ACKNOWLEDGEMENT

First of all, I am most grateful to the Almighty God for seeing me through my project work successfully.

I would like to thank my siblings Reuben Pometsey, Evans Pometsey, Emmanuel Pometsey and Felix Amenyo, they have been a source of encouragement and support to me throughout my studies and I am very grateful.

I would like to thank Dr. Joshua Dagadu, my supervisor and a lecture for his intense supervision, suggestions and inspiration which helped in the project development “I say thank you Doctor”.

I also want to thank all the lectures at the Information and Technology Education Department for the love and dedication they have for the job they do of which I am a beneficiary.

I am grateful to all my mates for the love and support they gave me during my study.

Lastly, to all who played a role in my life in diverse way, I really appreciate you and may the good Lord bless you abundantly.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In today's fast-paced world, customers demand swift and convenient services, making it imperative for businesses to offer a friendly and user-friendly system to expedite the checkout process. Traditional cash registers have been largely replaced by Point-of-Sale Systems (POS Systems), which combine both software and hardware to record sales in real-time. However, with the advent of technology and artificial intelligence, there is a compelling opportunity to enhance the efficiency of POS systems even further. Speech Recognition (SR) and Natural Language Processing (NLP), which fall under the domain of Artificial Intelligence, facilitate human-computer communication by enabling machines to understand and transcribe spoken language. This breakthrough offers an alternative input method to conventional keyboards and other hardware components. This thesis project aims to implement speech recognition technology in Point-of-Sale systems, reducing the reliance on traditional keyboard input method. The study poses two key research questions: whether Voice/Speech Recognition can effectively replace keyboard input in POS systems and how Voice/Speech Recognition can enhance service delivery in such systems. The research objectives include proposing speech recognition as a viable substitute for keyboard input and examining its impact on service delivery in POS systems. It addresses the practical application of speech recognition in point-of-sale systems using existing algorithms use in PHP, JavaScript (Ajax, jQuery) and MySQL with the help of Web Speech API, offering a novel approach to improve database interactions. In conclusion, this thesis explores the integration of speech recognition in Point-of-Sale systems as a means to optimize productivity and customer service, ultimately offering a valuable alternative to traditional input method

CHAPTER ONE

INTRODUCTION

1.1 Background to the study

A Point of Sale (POS) is a transaction where a customer exchanges money for goods or services from a merchant. It can also be defined as when and where a retail transaction is completed. Due to our tight schedules, we are always on the move. Human beings require faster services. Long queues at checkout often frustrate customers so a friendly user system will give practical help to cashiers. POS Systems became a saviour as it provides a faster way of doing business and a quicker checkout than cash registers (Kabir & Han, 2016). A Point-of-Sale System (POSS) is a computer software and hardware network that records sales as they occur. POSS refers to the recording of data and payment information of a customer at a physical sales point when goods or services are bought and sold (Dunn et al., 2016). Today's technology uses a barcode scanner and keyboard to retrieve the product information from the database, thereby making the checkout experience faster and ensuring that customers leave satisfied with the wish to return (Soto-Acosta et al., n.d.). A retail point of sales system has all the items registered into a database with product names and unique IDs. At the sales point, the customer asks for the things to buy, and the POSS user searches the database for the items. The transaction is done in many ways. Below are two examples.

A customer selects all needed products from the shelf and proceeds to pay and get a receipt.

In the second method, the customer pays and gets a printed receipt from the POS system desk containing all items to purchase. The receipt is then given to one of the shop attendees to select the items from the shelf.

The search process is done mainly in two ways. In example (a) above, the barcode on the items is shown to the barcode scanner device attached to the POSS hardware component, and the barcode id on the item is scanned. The captured id is then converted into text and printed into a text field or a text box. Based on the input text string, a search algorithm is performed, all information from the database about that particular product is retrieved, and the necessary transaction is done. In method (b), the item's name is typed with a keyboard into the search box, and, based on the input text string, the item is then retrieved, and the necessary transactions are performed. Software usability depends on how a system accomplishes an input task and how users support the system functions (Kabir & Han, 2016). With the increasing number of people accessing services from supermarkets simultaneously and the growing competition in the business environment, it is essential to ensure software quality (Kabir & Han, 2016).

Technology today has advanced at an incredible speed with many innovations. We now have self-driven cars through the introduction of Artificial Intelligence (AI). The main idea behind AI is to make work easier and maximize productivity. Speech Recognition (SR) and Natural Language Processing (NLP) is a branch of Artificial Intelligence (AI) that enables communication between humans and computers. In other words, NLP makes natural language understandable to machines and computers (Rao et al., 2010). SR can also be defined as a technology that can make machines recognize words spoken by humans through a microphone and convert them into text.(Joshi & Srivastava, 2015).

Just like the manipulation of data using keyboards, barcode scanners, magnetic card readers, mice, and many more as input for applications on computers and phones, SR also provides us with the ability to use voice as an input by the use of a microphone and a speech recognition engine (Joshi & Srivastava, 2015). The work of the speech recognition engine is to process and transcribe the voice input through the microphone into a text format that can be understood by the application involved.

This field of Artificial Intelligence has proven to be very difficult. Despite the difficulties, some achievement has been made. (Rao et al., 2010). Now, we have applications like SIRI, Amazon Alexa, and many others that implement speech recognition algorithms. Instead of typing on a keyboard to search for items on the internet, the user only speaks, and the search is done.

POS systems such as Square, Revel System, Elo's touchscreen, Oracle Micros, Toast, etc., have integrated speech recognition mostly for placing orders and payments but not for database interactions.

The purpose of this project is to implement speech recognition in Point-of-Sale systems to help make searching products from the database more accessible and faster and also as an alternative method to the use of a keyboard, barcode readers, magnetic card readers, chip readers, RFID readers/transmitters, etc. to increase productivity.

1.2 Statement of the Problem

Research shows that the average typing speed with a keyboard is about 40-60 words per minute compared to 80 to 100 words per minute with voice dictation tools (voxpow, 2020). SIRI, Cortana, and Amazon Alexa are a few on the list. Another report also indicates that speech dictation is three

times faster than typing on an iPhone keyboard (WEINER, 2016). The business world today is highly competitive. Long queues negatively affect customer service appreciation, therefore, a significant issue for companies (Bielen, 2007). Naidu also supports this statement: "Bottlenecks, long queue lines, and unnecessary waiting time is always associated with low productivity" (Naidu, 2009). No company can afford to lose a potential consumer due to operational delay in the supply of goods and services (Obi, 2014)

It has also been established that continuous keyboard use poses negative health issues (Toosi et al., 2011). It is laudable to say that implementing speech recognition in Point-of-Sale Systems will reduce long queues, improve customer satisfaction, and increase productivity because it reduces waiting time and provides quick checkout time. It also reduces keyboard risk factors such as Carpel Tunnel Syndrome and Tennis elbow, which affects heavy computer keyboard users.(Toosi et al., 2011). As no more stress is exerted on the wrist, there will be no more repetitive strain injuries (askergoworks, 2020). Using a keyboard also requires trained personnel who can type faster to provide quick service to customers. The existing system also requires the user to sit and engage both hands simultaneously. Still, with the implementation of SR, the user can search the database even if their hands are engaged temporally and can stand while doing the retrieval.

1.3 Research Questions

1. Can Voice/Speech Recognition be effectively used as a substitute to keyboard input in Point-Of-Sale (POS) systems?
2. How can Voice/Speech Recognition in Point-Of-Sale (POS) systems improve service delivery?

1.4 Research Objectives

The study examines how well speech recognition can be implemented in Point-Of-Sale (POS) systems to maximize productivity. The specific goals include the following;

1. Proposing voice/speech recognition as a substitute for keyboard inputs in Point-of-Sale (POS) systems.
2. To examine how Voice/Speech Recognition can improve service delivery in Point-Of-Sale (POS) systems.

1.5 Justification of the Study

Most of the literature on database interaction with speech recognition is focused on translating natural language into an SQL statement for non-technical users or users with little or no knowledge of SQL (Giordani & Moschitti, 2009). It requires that the user issue a command statement in a natural language (for example, find all students' names). After clicking a button, this statement is converted into an SQL statement (for example, Select * from students) (Deshpande & Devale, 2012). Even though this method has added to knowledge and broadened the capabilities of AI in natural language processing, there needs to be more literature on its implementation into existing systems. Moreover, most non-technical users interact with databases mainly through front-end systems written in object-oriented programming languages such as Java, C#, PHP, etc., which already have SQL statements embedded in the codes and, therefore, do not need to issue commands that will be translated to SQL statements. This study aims to fill the gap in knowledge by using speech recognition in a point-of-sale system to search for products in a database based on voice

input through a microphone. The technique is not limited to search but can be used for all SQL statements.

1.6 Scope of the Study

The study focuses on searching and retrieving product information from the database using voice instead of typing and other input methods. Existing algorithms provided by Web Speech Recognition API, PHP, Ajax, jQuery, and Structured Query Language (SQL) are used to successfully integrate Automatic Speech Recognition for retrieving items or products from the database to use in Point-of-Sale Systems. The work depends on the existing English grammar provided by Web Speech Recognition API.

1.7 Limitation of the Study

The study will not cover lexical, syntax, and semantic analysis and their implementation. It will also not cover language training. Users of this method must use English as the medium of communication.

1.8 General Layout of the Report

The study will consist of five main chapters. The first chapter is the introduction, which deals with the background of the study, problem statement, research questions, research objectives, the scope of the study, justification of the study, the research methodology, the limitation of the study, delimitation, and the organization of the study. The second chapter dwells on reviewing the literature related to the research study. Chapter three focuses on the methodology of the study, which entails the following: the research design, the research instruments, the validity and

reliability of data, and the data analysis procedure. The fourth chapter centers on the presentation of the results or findings of the study, the analysis of the results, and the discussion of the results. Chapter Five also deals with the reflection conclusion, summary, study recommendation, and suggestions for future studies.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview of NLIDBS (Natural Language Interface to Database System)

Natural Language Interface to Database System's primary purpose is to accept input mainly in English or any other natural language and translate it into a database query (B.Sujatha, 2012). Below are some reasons behind NLIDBS:

-To allow easy access to a database.

-No need-to-know SQL query language to access a database.

-Answer questions about the contents of the text.

-No need to know a database schema to interact with the database. The idea of NLIDBS started in the late sixties (Gaikwad, 2008), and research into its full implementation is still ongoing. The earlier systems that implemented the NLIDBS used the keyboard as the input method. One of the best-known NLIDBS in the sixties is LUNAR (Kaplan & Webber, n.d.). Lunar is a research prototype of a system built to make machines understand the natural English language rather than man adapting to machine language (Androutsopoulos et al., 1995). This system was built to analyse the chemical composition of lunar rocks. The systems of this period were built tailored to each individual database and, therefore, very difficult to implement or modify to be used in other databases and therefore issue of portability became a problem (Fin M & González Gutiérrez Dirigido por José Francisco QUESADA, 2019). A second period that extended to the late seventies saw the appearance of new techniques for NLIDBs. Several systems emerged that tried to improve

the previous techniques during this time. RENDEZVOUS (Templeton & Burger, n.d.) engage the users in dialogues to fulfill their requests. It was the first system to engage users in interaction. It uses dialogic features to improve results. Another unique system was the LADDER (Hendrix et al., n.d.). It was developed as an NLIDB of information about US Navy ships and used semantic grammar, "a technique that interleaves syntactic and semantic processing" (Androutsopoulos et al., 1995). It could produce impressive results but could have been more easily portable to other databases. Other systems of this period were PLANES and PHILIQA1 (Templeton & Burger, n.d.). This second period focused on improving the linguistic and conceptual analysis of queries but failed to face problems related to portability. The system CHAT-80 (Warren & Pereira, 1982) also came in the eighties. It was executed in “prolog.” Prolog is a high-level computer programming language first devised for artificial intelligence applications. According to (Warren & Pereira, 1982), the database of CHAT-80 consists of verities about 150 countries of the world and a small set of English languages that are enough for querying the database. It is designed to translate the English language question by making a logical form as the process of 3 serial and complementary processes. Logical constraints designate the first word. In the second process, words, nouns, and adjectives with their associated preposition are represented by predicates. The third process is a representation of a phrase by the conjunction of predicates. The procedures are the following; Parsing, Interpretation, Scoping.

The parsing module procedure determines the grammatical form of a sentence, and understanding and scoping consist of various translation rules expressed literally as PROLOG clauses. The primary approach pursued by CHAT-80 is to append some additional control information to the logical structure of a query to make it an efficient component of the PROLOG program, which can run directly to produce the answer.

Another program using this method was MASQUE (Androutsopoulos et al., 1995). These types of systems demonstrated the technique of using intermediate models to dissociate a query's linguistic expression and its semantic representation, in this case, in the form of logical expressions. In the mid-eighties, NLIDBs were a trendy area of research, and countless prototype designs were implemented. Much of that time's research was committed to portability issues. Some Systems that emerged in the mid-eighties were TEAM (Gross Barbara J et al., 1986), ASK (Henisz Thompson & Thompson, n.d.), JANUS (Rosnik Philip, 1989), EUFID (Burger, n.d.), and many others.

By this time, NLIDBs had become a hot topic in academia. Companies trying to produce commercially viable products using this technology followed suit. Some of these products available at the time were INTELLECT from Trinzic (Harris, 1984), Parlance from BBN, LanguageAccess (IBM's take in this endeavour), also Q&A from Symantec, Natural Language from Natural Language Inc., Loqui from BIM, English Wizard from Linguistic Technology Corp., among others (Fin M & González Gutiérrez Dirigido por José Francisco QUESADA, 2019). Despite the countless endeavours to build commercially practicable systems and the anticipations of rapid adoption, the topic started to relegate to the theoretical sphere after some time. Companies diverted their attention and efforts to other products, and database technologies pursued a different bath.

Listed in table 2.1 below is the overview of nine early NLDBS systems.

Year	Name	Domain	Language	Back-end	Technical approach
1973	LUNAR	Moon rocks	English	Database-specific	Grammar
1974	RENDEZVOUS	Airport flights	English	Database-specific	grammar, dialogue
1977	PHILIQA	General	English	heterogeneous	grammar, dialogue
1978	LADDER	US-Navy Ships	English	SQL	Grammar
1980	CHAT-80	General	English	Prolog	grammar, intermediate representation
1983	TEAM	General	English	Simple Object Database Access	semantic grammar, intermediate representation
1983	ASK	General	English	heterogeneous	intermediate representation
1983	EUFID	General	English	heterogeneous	semantic grammar, intermediate representation
1989	JANUS	General	English	Heterogeneous	intermediate representation

Table 1*:* Overview of some of the early NLIDB systems

In 2004, Ana-Maria Popescu, Alex Armanasu, Oren Etzioni, David Ko, and Alexander Yates developed PRECISE at the University of Washington. This development also explores the natural languages easily transformed into database queries. The preferred database is in the form of a relational database utilizing SQL as the query language. It presents the idea of semantically tractable sentences that translate to a distinctive semantic interpretation by analyzing some lexicons and semantic constraints. The system excels in processing the semantically tractable sentences it studies, but it could perform better with other types of queries and particularly needs help with nested structures.

NALIX, which Stands for Natural Language Interface for an XML Database, is an NLIDB system invented at the University of Michigan. Ann Arbor, Yunyao Li, Huahai Yang, and H. V. Jagadish developed this system in 2006. It used extensible markup language (XML) as a database with Schema- which aimed at retrieving data represented in XML format. It is a syntax-based system that produces XQuery words for XML documents. Its architecture includes three modules: a parse tree generator, a parse tree validator and a XQuery translation module.

XQuery is a query language developed specifically for retrieving information in XML. The concept is to use a keyword search for databases. However, pure keyword search clearly cannot be used. Therefore, some more affluent query mechanisms are added. XML elements relate to a collection of given keywords with several candidates.

2.2 State of the art of NLIDS

Also, concentrating on interactivity, we can cite NaLIR, which stands for "Natural Language Interface for Relational databases". (Li & Jagadish, 2014). It is an evolution of NaLIX, as explained earlier. Rather

than producing XQuery queries from natural language, this new version generates SQL queries. In this system, the authors provide an interactive environment where, with "a carefully limited interaction with the user, "they can answer complex user questions. At the same time, the system can provide a more interactive experience. The modelling of the queries (a tree structure) allows it to explain the results (or partial results) obtained during the dialogue. This is an exciting feature regarding interactivity and, for the sake of explainable artificial intelligence. This feature is becoming more demanding as AI technology spreads and its impact on society widens.

The work of Deshpande et al. engages the user to perform some tasks by engaging with a user interface. (Deshpande & Devale, 2012). This system has a user interface with five buttons arranged in order of execution. At first, the user clicks on the Manage Corpus button to load all the data dictionaries. Secondly, the user must enter Probabilistic Context Free Grammar after clicking on the second button, the PCFG Rules Setting. After this step, the user enters the query in a natural language in a text field or a text area. When the third button, the parser, is clicked, a parser tree is generated, and the PCFG Rule is applied by clicking the fourth button (PCFG Rule used button). The last button generates the SQL statement from the input English language and displays it in the second text field or text area.

Kaur et al.'s work is similar to that of Deshpande et al. However, this user interface has only one button. The matching of tables and attribute names is done after the Natural Language Processing (NLP) is done on the user input using a mathematical model called Automata. (Kaur et al., 2013). The user clicks on the button on the user interface to convert the natural input language into an SQL statement and display it on the screen. Other works by (Rao et al., 2010,Gaikwad, 2008,Saravjeet Kaur & Rashmeet Singh Bali, 2012) all engage users with user interfaces

developed to accept inputs in natural language, mainly in English and convert them into SQL statements.

2.3 End-to-End Speech Recognition

End-to-end speech recognition has revolutionized the traditional pipelines by directly mapping input audio to transcriptions, making the process more streamlined. The Listen, Attend, and Spell (LAS) model, introduced by Chan et al. (2016), is an excellent example of this approach. Using recurrent neural networks (RNNs) with attention mechanisms has helped LAS achieve state-of-the-art performance, leading to further research in end-to-end architectures. The Deep Speech models by Hannun et al. (2014, 2017) are also noteworthy, as they employ convolutional neural networks (CNNs) and bidirectional RNNs to transcribe audio, eliminating the need for handcrafted features. The Deep Speech architectures are popular and impactful because of their simplicity. The Connectionist Temporal Classification (CTC) loss function is a significant factor in the success of end-to-end approaches for large-scale applications. Graves et al. (2006) introduced CTC, eliminating the need for explicit alignments between input and output sequences and facilitating end-to-end learning. The approach of training models for speech utterances of different lengths has been beneficial. The latest advancements in this field include Transformer-based models like the Conformer architecture (Gulati et al., 2020). Conformers utilize self-attention mechanisms, which offer parallelisation benefits and improved modelling of long-range dependencies in speech signals. Overall, the evolution of end-to-end speech recognition, from LAS and Deep Speech to CTC-based models and Transformers, has transformed the field of automatic speech recognition. These architectures directly map audio to transcriptions and can create more efficient, scalable, and accurate speech recognition systems.

2.4 Speech Dataset

Speech datasets are essential for developing automatic speech recognition (ASR) and related fields. LibriSpeech, a corpus created from public-domain audiobooks, has been instrumental in training large-scale multilingual ASR systems (Panayotov et al., 2015). The Mozilla Common Voice project has also contributed to the advancement of the field by providing a linguistically diverse dataset, which fosters research in multilingual ASR applications (Laurenzano et al., 2020). In speaker recognition, the VoxCeleb dataset, which features diverse collections of celebrity speech samples, serves as a benchmark for evaluating speaker recognition technologies (Chung et al., 2019). The TIMIT Acoustic-Phonetic Continuous Speech Corpus, a collaboration between Texas Instruments and MIT, is still foundational for phonetic research (Texas Instruments & MIT, n.d.). Sak et al.'s (2014) study of bidirectional LSTM RNNs in acoustic modelling has significantly influenced the adoption of deep learning architectures in speech recognition. This comprehensive study sheds light on the effectiveness of such models. Addressing the demand for limited-vocabulary datasets, the Speech Commands dataset introduced by Warden (2018) has played a crucial role. With a focused vocabulary, this dataset facilitates the development of models tailored for specific applications like voice-activated command systems. In summary, the evolution of speech datasets, exemplified by LibriSpeech, Mozilla Common Voice, VoxCeleb, TIMIT, and Speech Commands, has propelled research in ASR, speaker recognition, and deep learning applications within the field.

2.5 ASR (Automatic Speech Recognition)

Over the recent years, we saw a progressive advancement and growth of Automatic Speech Recognition (ASR) technologies (Yu & Deng, n.d., Ravanelli et al., 2017), which have reached outstanding performance levels and are used by millions of people worldwide nowadays. Deep learning plays a vital role in this technological breakthrough (Kim, 2016), overcoming earlier speech recognizers founded on Gaussian Mixture Models (GMMs). Above deep learning, other factors have played a role in the progress of the field. Several speech-related tasks, such as AMI (Renals et al., n.d.) and DIRHA (Cristoforetti et al., n.d.) and speech recognition challenges, such as CHiME (Barker et al., n.d.), Babel, and Aspire, have significantly boosted the improvement in ASR. The general distribution of large datasets such as Librispeech (Panayotov et al., n.d.) has also been essential in establishing standard evaluation framework works and tasks. Among the other factors, the growth of open-source software such as The Hidden Markov Model Toolkit (HTK) (Young, 1994), Julius (Lee & Kawahara, n.d.), CMU-Sphinx, RWTH-ASR (D. Rybach et al., n.d.), LIA-ASR (Linarès et al., 2007) and, more recently, the Kaldi toolkit (Povey et al., n.d.) have also helped popularize ASR, creating both research and development of novel ASR applications significantly more available. Kaldi currently designates the most famous ASR toolkit. It relies on finite-state transducers (FSTs) (Mohri, n.d.) and provides a collection of C++ libraries for efficiently executing state-of-the-art speech recognition systems.

CNTK (Microsoft's open-source deep-learning toolkit) (Temiz, n.d.), have attained popularity in the machine learning community. These toolkits offer tremendous flexibility in neural network design and are used for various deep-learning applications. PyTorch (Paszke et al., n.d.) is an

emerging python technology that enforces efficient GPU-based tensor analyses and enables the design of neural architectures. An exciting component of PyTorch lies in its current and adaptable design that innately supports dynamic neural networks. The computational graph is dynamically created on the fly at a run time instead of statically compiling. The PyTorch-Kaldi project seeks to bridge the gap between Kaldi and PyTorch1. Nevertheless, my focus will be on Web Speech API for this project. The W3C Speech API Community Group develops the Web Speech API. It is a recent initiative, and the specification draft awaiting final commitment was published only in October 2012. Work on the Web Speech API can be traced back to the end of 2011. The Web Speech API covers speech analysis and synthesis (Adorf, 2013). In other words, it supports the conversion of speech to text and vice versa. The API is developed purely in JavaScript, which is currently one of the prevailing client-side scripting languages of the web. The Web Speech API is event-based, which fits nicely with the coding style of JavaScript. The user agent handles calls to the API, which takes charge of all communication with a web-based speech recognition service. It requires that the user agent implements the API. The event-based architecture allows programs to process speech asynchronously (Adorf, 2013). Events are used to report intermediate speech recognition results, which is convenient because it allows programs to give almost immediate feedback to the user. Speech recognition can be interrupted at any time, which is convenient because it relieves the web developer of extra work in the event handler routines. The API supports many languages. The user specifies which language to use before speech recognition starts. The API defines ways to adapt the speech recognition system for specific tasks. Grammar can be given to the speech recognition system. The system can take advantage of the constraints given by the grammar for improved speech recognition. It is written in the specification that the grammar format is still subject to discussion and still needs to be finished. Intermediate or final recognition results

are given in several candidate sentences, each associated with a specific confidence value. The most likely transcription is listed first. The API distinguishes between parts of a transcription that are preliminary and final parts (Adorf, 2013). It is useful when looking at intermediate results while speech recognition is ongoing.

Five popular browsers are currently used, including Google Chrome, Mozilla Firefox, Microsoft Internet Explorer, Safari, and Opera (w3schools.com, n.d.). However, only Google Chrome (version 25+) has experimental support for the Web Speech API.

2.6 ASR (Automatic Speech Recognition) and NLIDS (Natural Language Interface To Database System)

With the emergence of ASR projects and APIs, new possibilities are created for integrating ASR in NLIDBS. Researchers explore using voice input instead of traditional keyboard input, as in the past. Below is a review of a few examples.

Database Interaction Using Automatic Speech Recognition (Thansekhar & Balaji, n.d.) This work converts speech to text using Sphinx4 (agdy Bayoumi, 1994), an HMM-based speech recogniser (International Islamic University Malaysia Kulliyyah of Engineering et al., n.d.). The text is used as input. An administrator creates an XML configuration file from a created Database. Each configuration file line is processed, and the required details are taken to produce the output for text to SQL converter. To arrive at the SQL statement from the input text, parsing, tokenisation and other processes are performed as the other models discussed earlier. The work of (Mahajan et al., n.d.)also takes input from a microphone and uses pattern matching to construct the query from the user's voice input and provide output. Users interact with a provided web application. In addition

to the works mentioned above (Wahi et al., n.d.) also used Google Speech Recognition API to convert input speech into text which is also converted into token streams with the help of a tokeniser. After the input text goes through a series of processes, such as parsing with the help of Stanford Parser, OpenNLP and Keyword extraction Module, the final SQL statement is generated, and the final output is given to the user.

2.7 Challenges of NLIDS

A study has been done over the last few decades on Natural Language Interfaces. With improved hardware processing power, many NLIDBs cited in the historical background got favourable results. Although several NLIDB systems have also been developed so far for commercial use, the use of NLIDB systems is limited. It is not a standard alternative for interfacing with a database (Nihalani et al., 2011). This lack of endorsement is mainly due to the numerous shortcomings in the NLIDB system to understand a natural language. Issues such as scoping (determining which interpretation should be taken due to modifier attachment problems in natural language inputs) and ambiguous terms due to the use of conjunctions and disjunctions words. Other factors, such as false linguistic expectations, result from uncertainty in linguistic coverage. This problem stems from the fact that the linguistic coverage of any NLIDB system is limited (Fin M & González Gutiérrez Dirigido por José Francisco QUESADA, 2019).

2.8 Proposed Solution

As stated in the justification section in chapter 1 and the challenges section in chapter 2, the NLIDBS issues affect its integration into existing systems. To solve some of these issues, I propose a system that uses natural language input and existing methods to manipulate database systems

with the help of SQL queries. The main objective of NLDBS is to develop an interface for non-technical users of databases. However, recent development in programming languages has made it easy to integrate SQL queries into software programs used by non-technical users (with no or little knowledge of SQL). Such systems include School Management Systems, Hospital Management Systems and many more. Users of these systems are only trained to use them, not the technicalities behind them.

The proposed system is a web application that accepts user voice input from a microphone and transcribes the voice to text with the help of Web Speech API. PHP codes with embedded SQL script, JavaScript, Ajax and jQuery are used to retrieve information from the database based on the transcribed voice-to-text. The aim is to integrate this system into a Point-of-Sale system to retrieve product information and perform wildcard searches.

CHAPTER THREE

METHODOLOGY

3.1 Research Design

The study uses the Simulation Research Design method to demonstrate the feasibility of integrating speech/voice recognition as an alternative to keyboard input in point-of-sale systems. Five users of varying ages and levels of experience with technology will be selected to accomplish this goal. These participants will be asked to interact with a database using voice and keyboard inputs. Their performance will be measured based on task completion time, accuracy, and error rates. Data will be analyzed and meaningful conclusions drawn based on the performance metrics.

3.2 Tools and Materials

Microsoft Windows 10 and above operating System, running the latest version of Google Chrome or Microsoft Edge. The minimum hardware specification is 4GB RAM and a processor speed of 1.8GHz. Open-source scripting languages will also be used, specifically PHP, JQuery, JavaScript, Ajax and SQL. MySQL will be used to manage the database, while Apache (XAMPP) serves as the local server for the simulation. To convert speech to text, the Web Speech API will be used. A fast and reliable internet connection is crucial for optimal performance.

3.2.1 PHP

It is a server-side scripting language specifically designed for web development. Its features make programming dynamic web applications easier (Suehring Steve & Valade Janet, n.d.). Rasmus

Lerdorf first created it in 1993 (Wikipedia, n.d.). You can embed dynamic activity by simply giving web pages a .php extension (Nixon, n.d.).

Here's how PHP works:

First, the web server scans the file in HTML mode, assuming the statements are HTML and sends them to the browser without processing. It continues in HTML mode until it encounters a PHP opening tag (<?php). At this point, the web server hands the processing to the PHP module, which executes the PHP statements. If there is output from PHP, the server sends it to the browser. The web server continues in PHP mode until it encounters a PHP closing tag (?>). When this happens, it returns to HTML mode and resumes scanning. With PHP, you can have complete control over the web server. You can even perform actions like adding, deleting, and fetching user details when used with the database. PHP is used in this study to retrieve product details from the database.

3.2.2 JavaScript

JavaScript was developed to allow for interactive and dynamic user experiences on HTML web pages. It was invented by Brendan Eich in 1995 (W3C, What is JavaScript, n.d.). It enables tasks such as validating email addresses on input forms and displaying error messages. JavaScript is a key component of modern web pages and is responsible for various widgets and usability features that users take for granted. When user types in a search field on a website, it suggests what to do, thanks to JavaScript. When combined with CSS, JavaScript powers dynamic web pages that update without requiring a new page to be loaded from the server. Unlike PHP, JavaScript programs run on the user's web browser, making it a client-side scripting language.

3.2.3 jQuery

jQuery is a tool that simplifies JavaScript programming by adding it to web pages. With jQuery, tasks such as using *'getElementById'* are made easier. It includes selectors that provide more powerful ways of accessing web page elements for use in JavaScript (Suehring Steve & Valade Janet, n.d.). Moreover, jQuery makes cross-browser development easier for JavaScript users. The original JavaScript support varies widely from browser to browser and version to version, which can cause compatibility issues. jQuery solves this problem by accurately identifying the browser being used and adjusting its functions accordingly to ensure consistent behaviour across browsers. The experiment made use of '*getElementById'* method for selecting items from HTML forms and therefore the need to use JavaScript and jQuery.

3.2.4 Ajax

The concept of "Ajax" was first introduced in 2005. It is short for Asynchronous JavaScript and XML, which essentially utilises certain JavaScript methods to transfer data between a browser and server in the background(Nixon, n.d.). A prime example of this technology is demonstrated by Google Maps, where new areas of a map are downloaded from the server as needed without requiring a page refresh (Nixon, n.d.).

AJAX employs a combination of the following:

• A browser's built-in XMLHttpRequest object (to request data from a web server)

• JavaScript and HTML DOM (to display or use the data)

With AJAX, web pages can be updated asynchronously by exchanging data with a web server behind the scenes. This implies that it is possible to update specific parts of a web page without reloading the entire page ((W3C, What is Ajax, n.d.)). The experiment used JavaScript-driven events called DOM (Document Object Model) to retrieve data from the database without a button click. To achieve this, a combination of jQuery and Ajax methods was employed.

3.2.5 MySQL

MySQL is a commonly used open-source Relational Database Management System (RDBMS) that can manage multiple databases simultaneously. It was designed to be fast and compact, making it ideal for website use. Data is stored in tables within the database, and these tables are organized into rows and columns. Each row represents an entity, such as a category or product, while each column contains specific information about that entity such as a category name, a product name etc. You can interact with the database using Structured Query Language (SQL), a standard computer language most database management systems understand. SQL is similar to English and can be used to create commands such as CREATE, DROP, ALTER, SHOW, INSERT, LOAD, SELECT, UPDATE, and DELETE. These commands can be used to retrieve specific information from the database, such as ‘’ SELECT * FROM products WHERE category=' Tomatoes';’’

The above SQL statement will retrieve all products under the category ‘Tomatoes’ that have already been inserted into the MySql database.

PHP can be used to make these calls directly to MySQL without the need for a command-line interface (Suehring Steve & Valade Janet, n.d.). The experiment involves the creation of a database and MySQL fits in because it is especially popular for use with PHP-based websites.

3.2.6 XAMPP

XAMPP is a free and open-source cross-platform web server. The term XAMPP is an acronym for Cross-Platform, Apache, MySQL, PHP, and Perl. This popular web server enables programmers to write and test their code on a local webserver. Apache Friends developed it and offers the public the ability to revise or modify its native source code. Additionally, it includes MariaDB, Apache HTTP Server, and interpreters for various computer languages such as PHP and Perl (educba, n.d.).

3.2.7 Web Speech Api

The Web Speech API, developed by the W3C Speech API Community Group, allows for speech analysis and synthesis (Adorf, 2013). It enables the conversion of speech to text and vice versa and was first introduced in 2011 before being published in October 2012. The API is written in JavaScript, one of the most commonly used client-side scripting languages for web development. Its event-based architecture is well-suited for JavaScript coding styles, with the user agent handling calls to the API and all communication with a web-based speech recognition service. The API supports many languages and allows for the adaptation of speech recognition systems for specific tasks. Interruptible speech recognition and the use of events to report intermediate results enable programs to process speech asynchronously and provide near-immediate feedback to users, without requiring extra work from web developers (Adorf, 2013). The API distinguishes between

parts of a transcription that are preliminary and final parts (Adorf, 2013). It is useful when looking at intermediate results while speech recognition is ongoing.

3.2.8 Sublime Text

Sublime Text is a versatile text and source code editor that is available as shareware for Windows, macOS, and Linux. It comes with built-in support for various programming languages and markup languages. Additionally, users can make it their own by customizing it with themes and expanding its functionality with plugins that are usually developed and maintained by the community under free-software licenses. The editor has a minimal interface and provides programmers with useful features such as configurable syntax highlighting, code folding, regular expression-based search-and-replace, a terminal output window, and more. Although it is proprietary software, a free evaluation version is offer (wikipedia, n.d.).

During the experiment, the text editor used for writing PHP and JavaScript codes was Sublime Text.

3.3 Experimentation

This section explores and evaluates the use of voice or speech for interacting with databases, with a focus on comparing voice commands with keyboard inputs. As we move towards a more digital world, new methods of human-computer interaction are emerging, and voice commands are becoming a popular alternative to traditional inputs. This experiment aims to demonstrate how voice or speech recognition can be integrated into structured database interactions. The project also aims to compare the performance of voice and keyboard commands under different conditions and usage scenarios, examining their effectiveness, response time, accuracy, and error rate.

3.3.1 Experimental Setup

To carry out the experiment, XAMPP was first installed and all the necessary configurations done.

After successful installation of XAMP the interface should look like the figure 3.1 below.

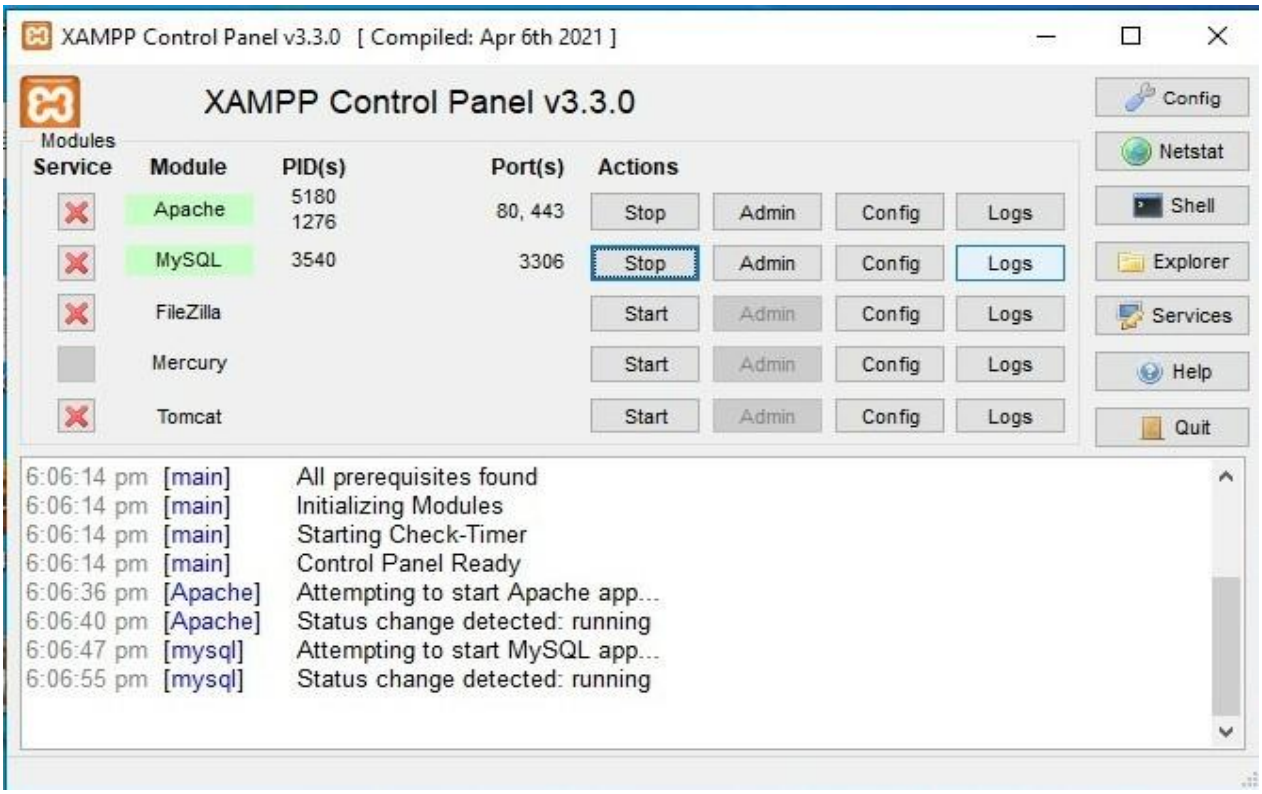


Figure 1: XAMPP Interface.

The second installed application is Sublime Text which was used throughout for coding and editing text for the experiment.

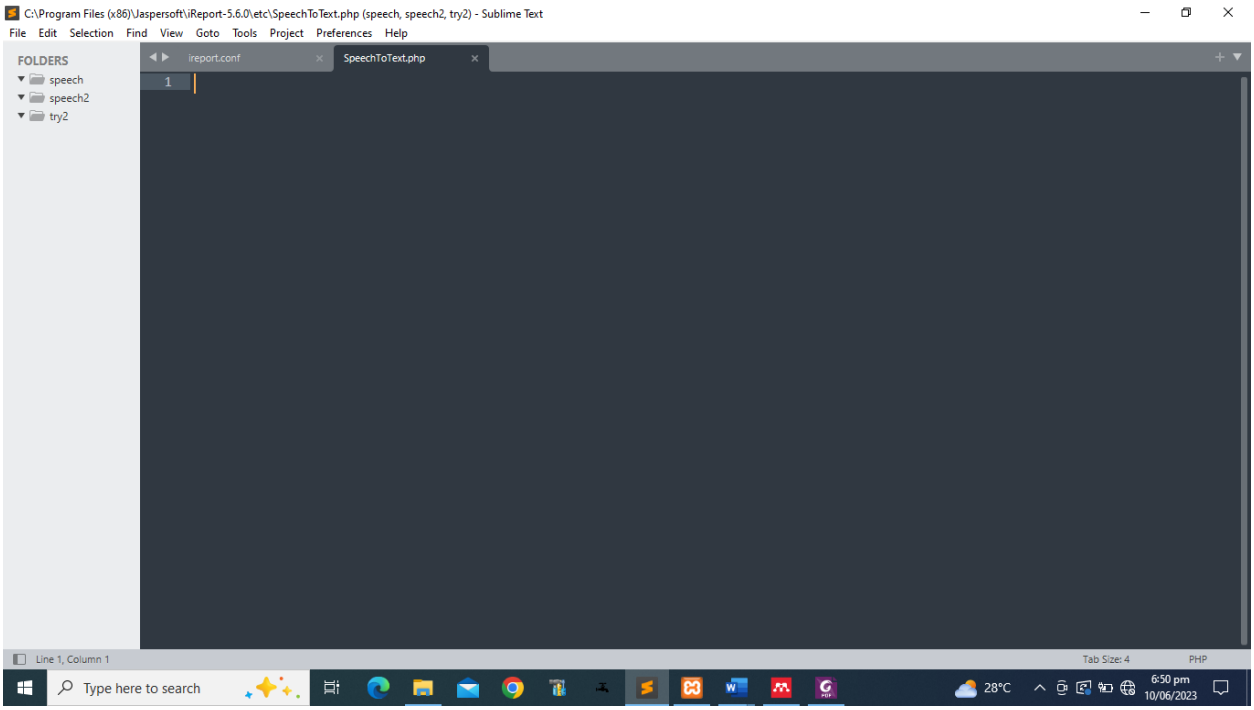


Figure 2: Sublime Text Interface

3.3.1.1 Database

In order to initiate the simulation, a database named "pos1_db" was established to contain the entities that will be exhibited. Two tables, "Category" and "new_product," were created within this database. The "new_product" table contains entities such as ProID, Productname, cost_price, Selling_price, and CatID, whereas the "Category" table includes CatID and Category. While a category can have numerous product names, a product name can only belong to one category

The screen-shot below shows the Database, table and the entities.

Figure 3:Database and Tables

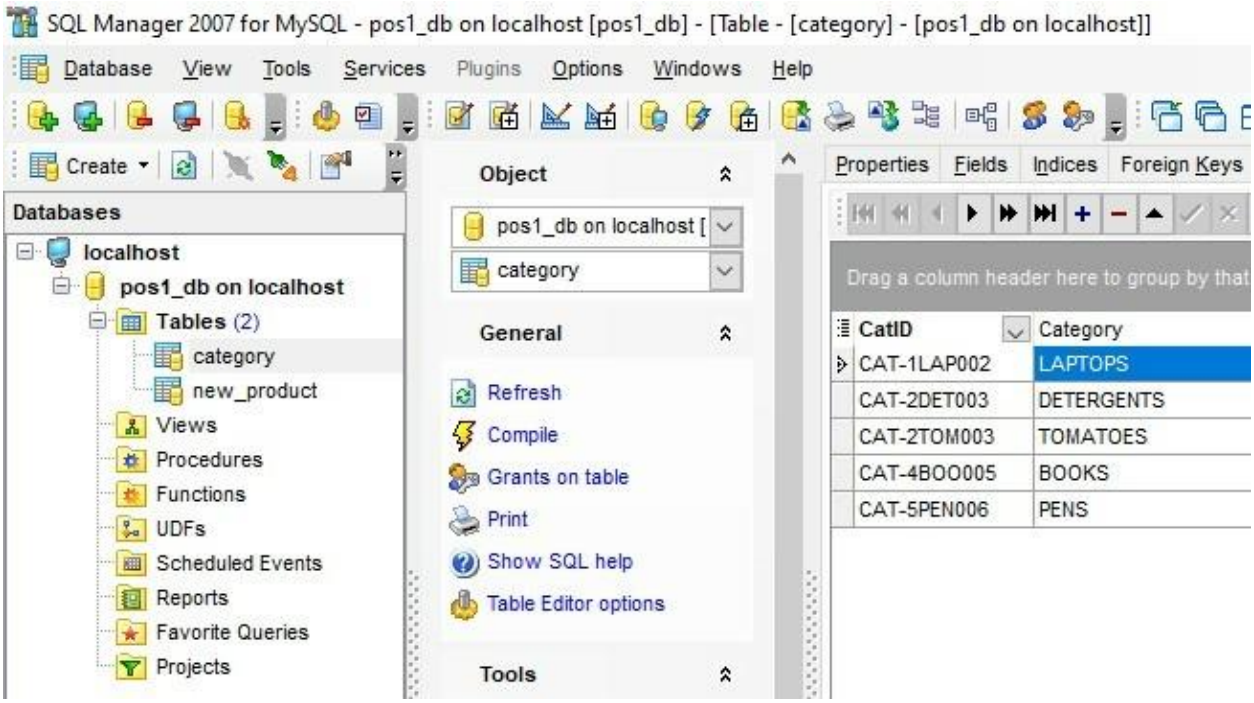
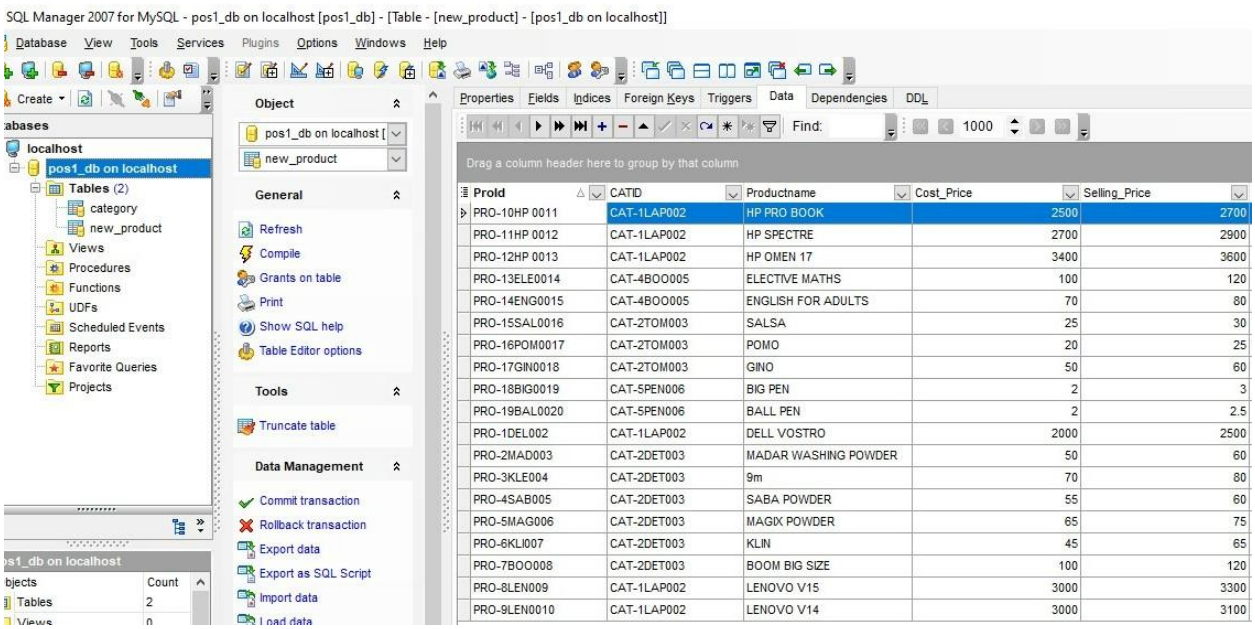


Figure 4:Entities in the Category table



Prold	CATID	Productname	Cost_Price	Selling_Price
PRO-10HP 0011	CAT-1LAP002	HP PRO BOOK	2500	2700
PRO-11HP 0012	CAT-1LAP002	HP SPECTRE	2700	2900
PRO-12HP 0013	CAT-1LAP002	HP OMEN 17	3400	3600
PRO-13ELE0014	CAT-4BOO005	ELECTIVE MATHS	100	120
PRO-14ENG0015	CAT-4BOO005	ENGLISH FOR ADULTS	70	80
PRO-15SAL0016	CAT-2TOM003	SALSA	25	30
PRO-16POM0017	CAT-2TOM003	POMO	20	25
PRO-17GIN0018	CAT-2TOM003	GINO	50	60
PRO-18BIG0019	CAT-5PEN006	BIG PEN	2	3
PRO-19BAL0020	CAT-5PEN006	BALL PEN	2	2.5
PRO-1DEL002	CAT-1LAP002	DELL VOSTRO	2000	2500
PRO-2MAD003	CAT-2DET003	MADAR WASHING POWDER	50	60
PRO-3KLE004	CAT-2DET003	9m	70	80
PRO-4SAB005	CAT-2DET003	SABA POWDER	55	60
PRO-5MAG006	CAT-2DET003	MAGIX POWDER	65	75
PRO-6KLI007	CAT-2DET003	KLIN	45	65
PRO-7BOO008	CAT-2DET003	BOOM BIG SIZE	100	120
PRO-8LEN009	CAT-1LAP002	LENOVO V15	3000	3300
PRO-9LEN0010	CAT-1LAP002	LENOVO V14	3000	3100

Figure 5:Entities in the new product table.

3.3.1.2 Proposed System Workflow

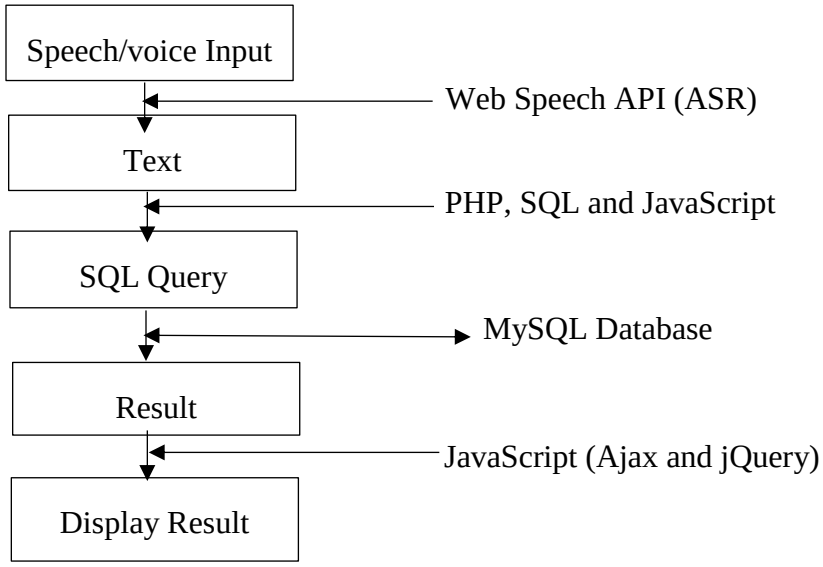


Figure 6: Diagram of Proposed System

3.3.1.3 Web Speech Api (ASR) Module

The system utilizes an automatic speech recognition feature that converts the user's speech/voice input into text output. This continuous speech recognition system works autonomously, regardless of the speaker. I implemented this module with the help of the Web Speech API tool, and to record user input, I used a microphone. I designed a textbox that features a microphone icon. This icon has two settings: On and Off. When clicked, the microphone initiates the voice input process, and the icon changes to display a strikethrough microphone. To stop the process, users can either click the strikethrough icon or give a voice command of "Stop".

3.3.1.4 Text to SQL and PHP Module

The input recognized in the Web Speech API module is converted into a standard text format in this module. The text format output is kept in a text box. A server-side scripting language, PHP is used with SQL and based on the transcribed voice-to-text output; an SQL statement is generated. For example:

$sql="SELECT * FROM new_product WHERE category LIKE '%'. $_POST["query"]."%'"; The variable “query” in the SQL statement above stores the value from the textbox, which contains the input text. With the help of JavaScript DOM events (onkeyup), the SQL command is fired to the database without a button click.

3.3.1.5 Database Interaction Module

In this module, we will be utilizing the SQL query obtained in the previous module to retrieve essential details from the database. The required information will be obtained once the query is

sent to the database. Although it is a standard database interaction, the query is not entered via a keyboard but through the user's voice input. The outcome will be displayed on the screen.

3.3.1.6 Display Module

The second module utilizes the jQuery library Ajax to showcase the SQL statement. Ajax functions and methods can load data from the server without refreshing the browser page. This technology presents a list of products based on their respective categories. If users select a product from the list, they can access additional information, such as the cost and selling prices, displayed in two separate text boxes. Alternatively, a user can mention a product from the list to retrieve its details without clicking on it.

3.3.2 Simulation

3.3.2.1 Process One (Wild Card Search Command)

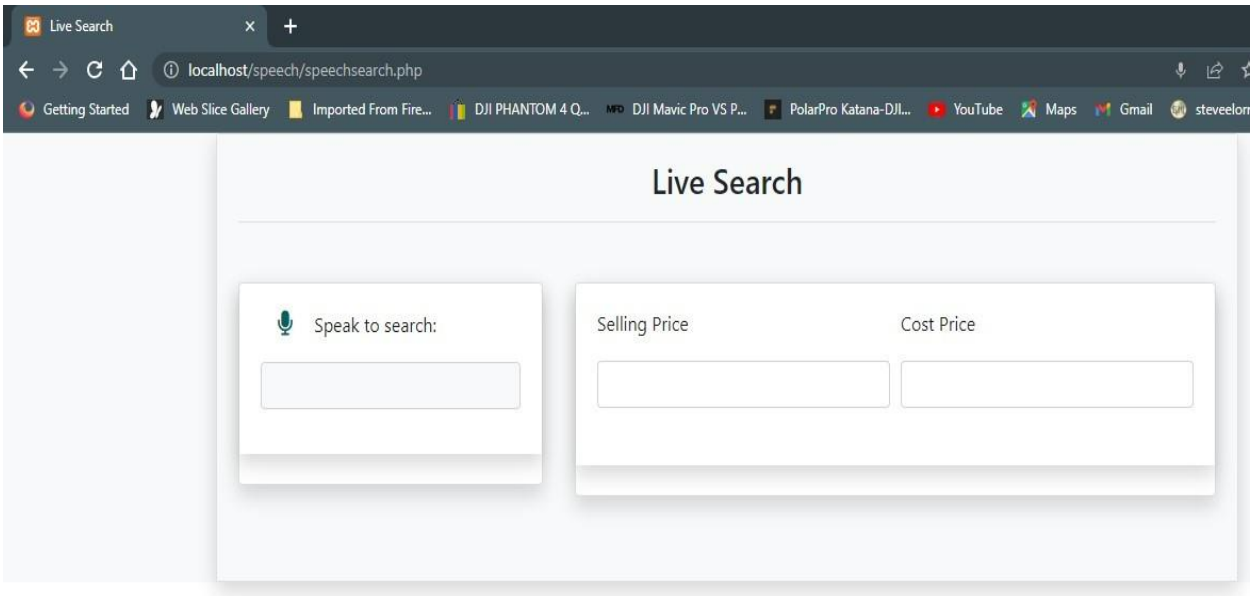


Figure 7:User Interface of proposed system

STEP 1

A user activates the process by clicking on the microphone icon on the interface shown in Figure 3.7 above. As soon as it is activated, a strikethrough icon is displayed and also a record icon is shown at the right side of the page title. The system is now ready to accept voice input. The output is displayed in Figure 8 below.

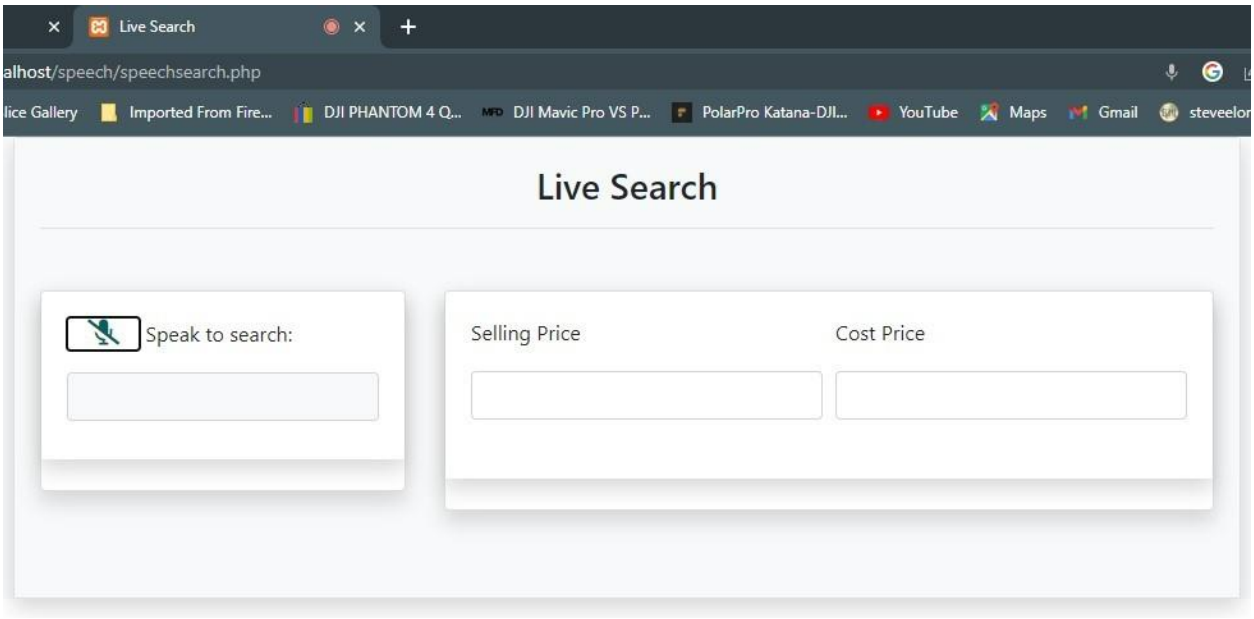


Figure 8: System ready for voice input

STEP 2

A user mentions a category name through an attached microphone, all items registered under that category in the database is populated in a list form. Example the user mentioned laptops as the category name and all items under laptops are displayed in a list.

Below is the output in Figure 9.

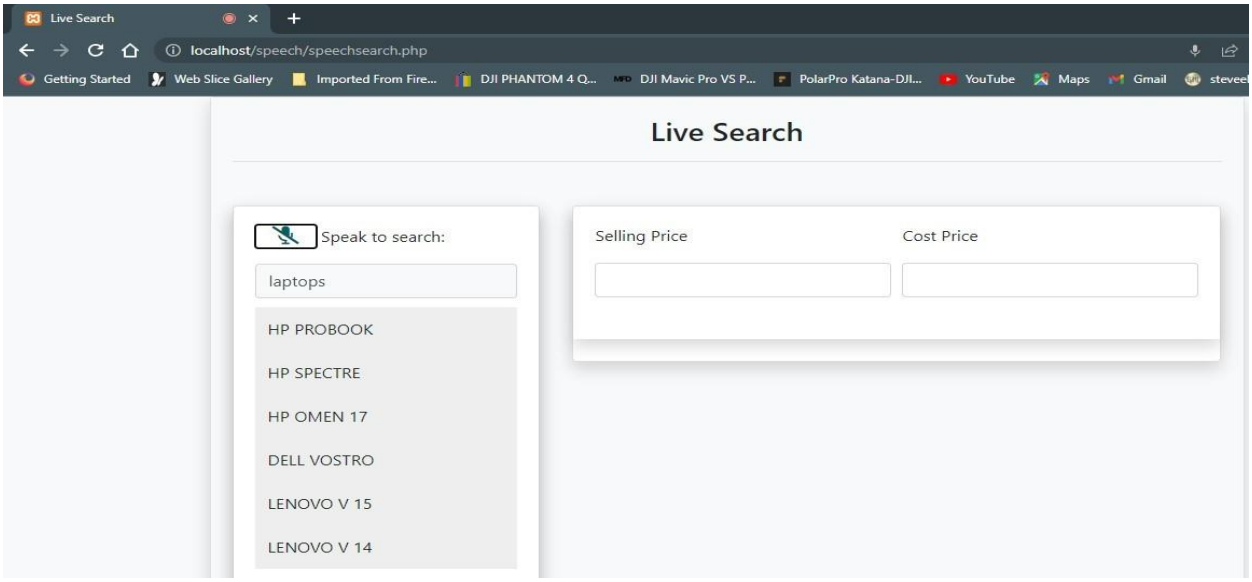


Figure 9:Category output.

STEP 3

The next step is to fetch information from the database on any of the items under this category. To do this a voice command “clean” is used to clear the text field. After that the user can go ahead and mention any of the products under this category. Below is the screenshot in Figure 10

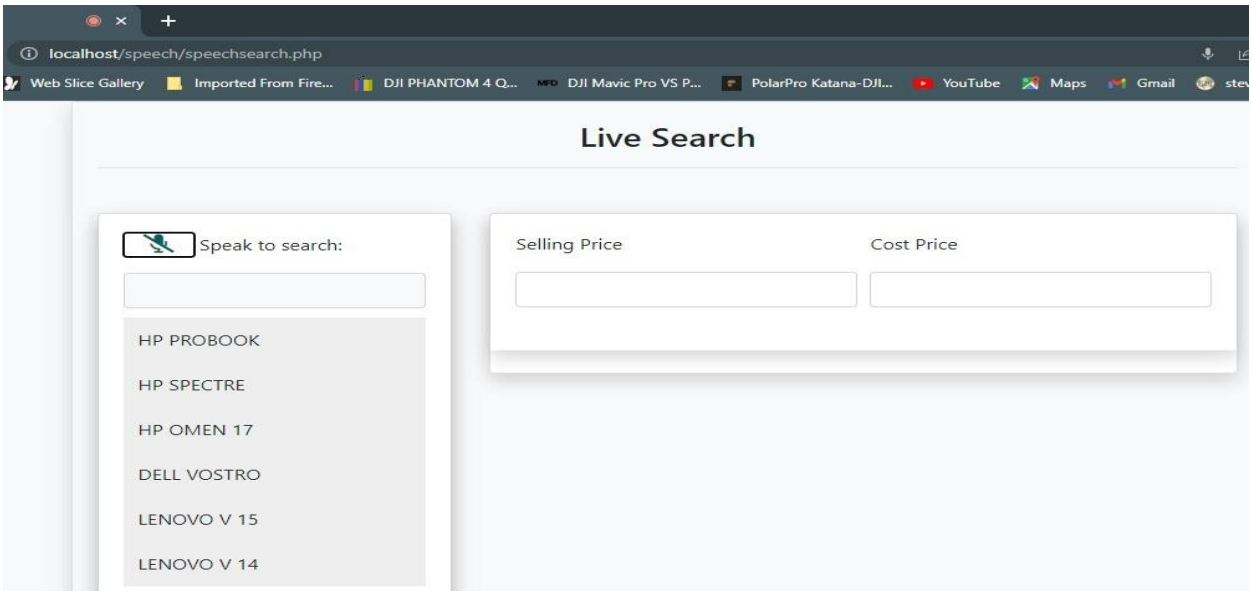


Figure 10:“Clean” command clears field.

STEP 4

The user can call a product name from the list and view its details in the text fields. Once the product name appears in the text field, and the user uses the "go" command, the selling and cost price of the product are displayed in the other two text fields. For instance, in the screenshot below, the user selected Dell Vostro and the selling and cost price were shown in the other two fields. Please refer to Figures 11 and 12 for the screenshots.

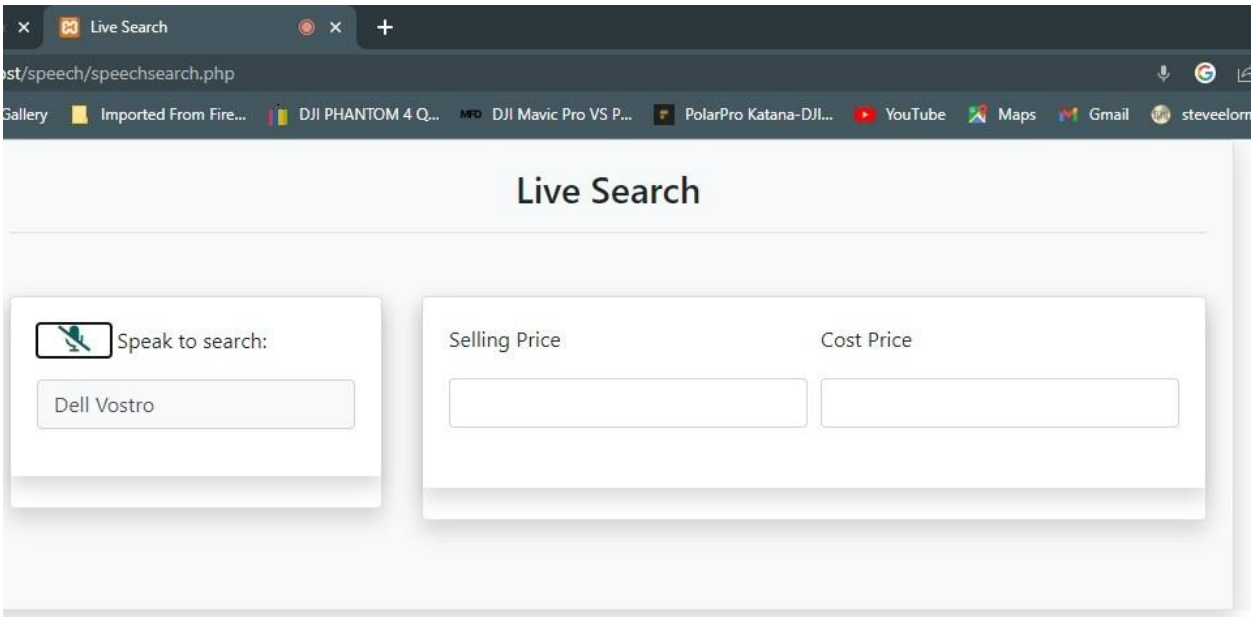


Figure 11:Voice selected product

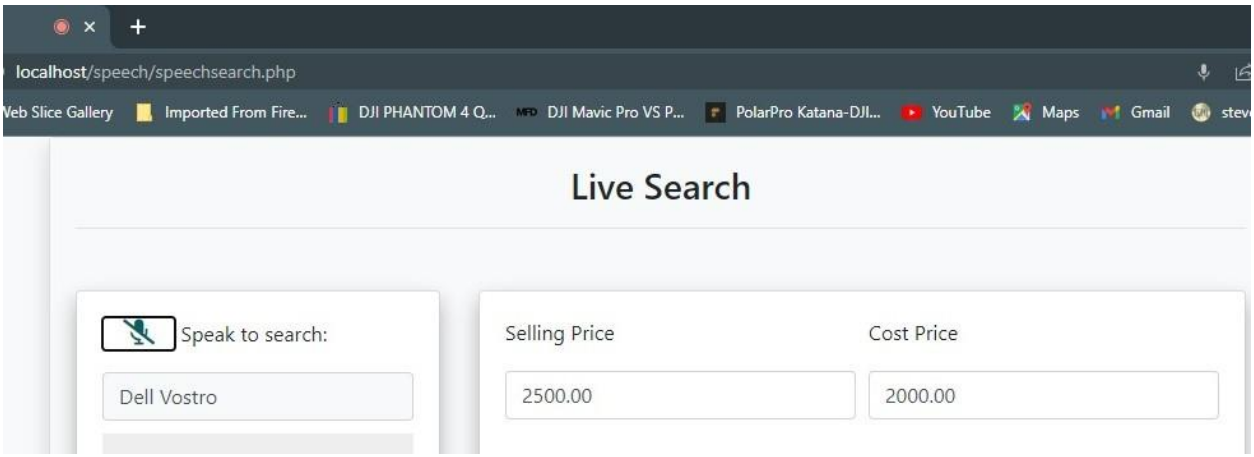


Figure 12:Further information on the product after the “Go” command

3.3.2.2 Process Two (Select Command)

STEP 1

As explained in 3.7 step 1, a user activates the process by clicking on the microphone icon on the interface shown in Figure 8 to activate the live select.

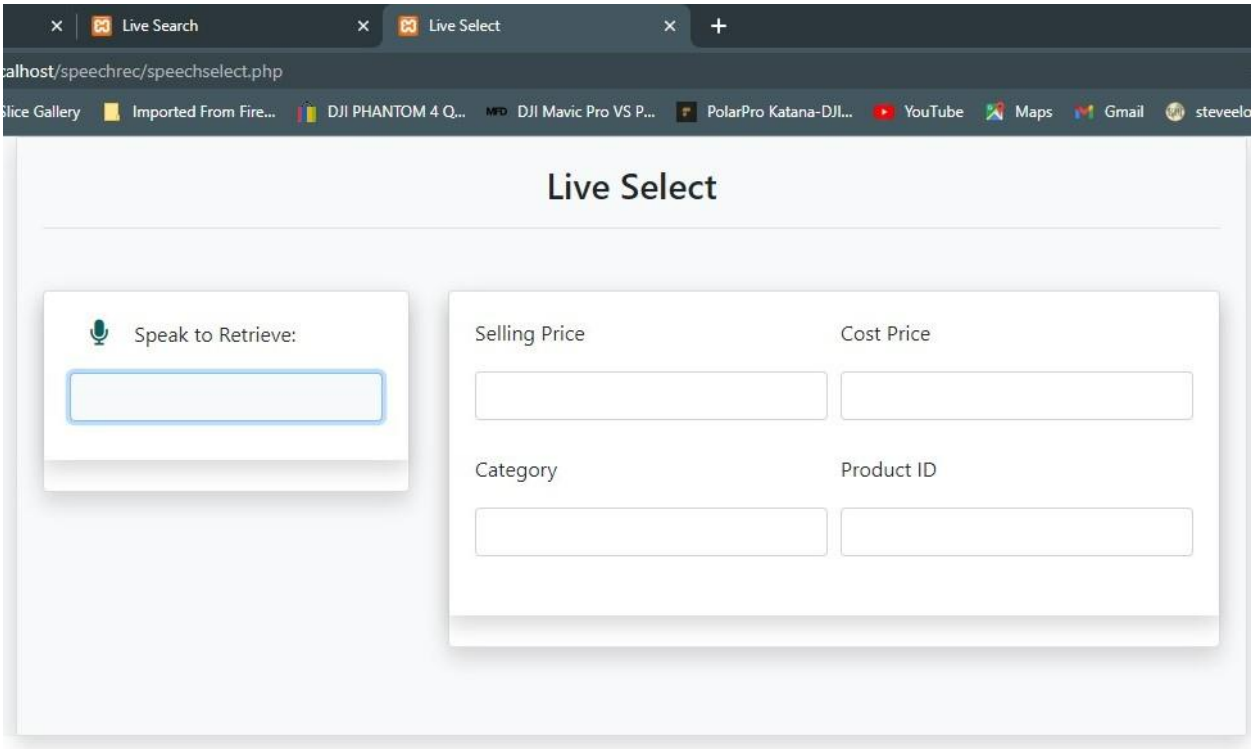


Figure 13: Select Interface

STEP 2

When the microphone is activated, the user only mentions the product name, and automatically, the selling price, cost price, product category and product ID are retrieved from the database and filled in the respective text fields. For example, in Figure 14 below, the user mentioned Dell Vostro and all the additional information is retrieved and shown in the respective text fields.

Below is the screenshot in Figure 14

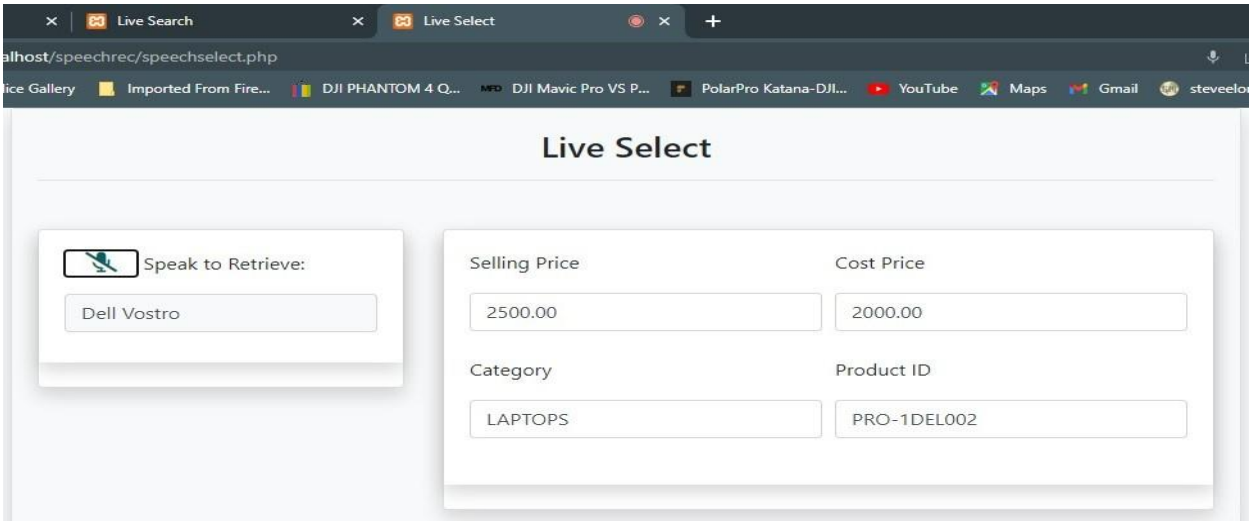


Figure 14:Retrieved Information 1

Another example is shown in Figure 15 below. The user mentioned elective maths.

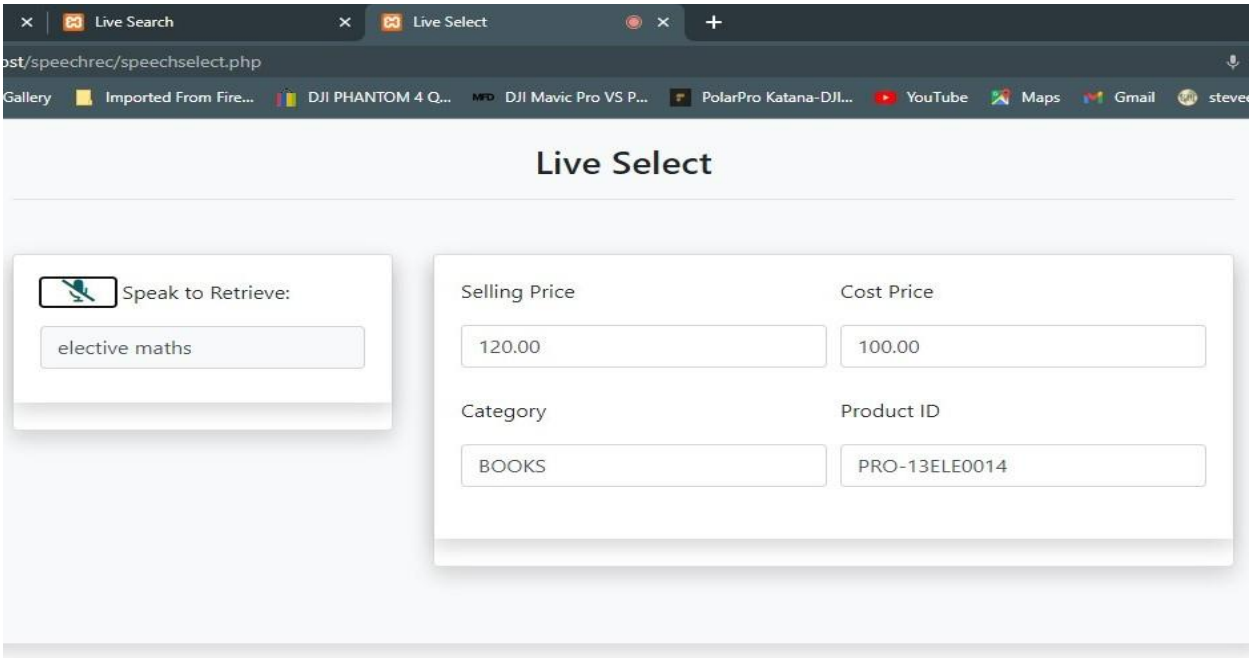


Figure 15: Retrieved Information 2

3.4 Evaluation Metrics

Comparing voice input and keyboard input for efficiency in a voice-controlled database involves evaluating various aspects of both methods. Here's a systematic approach to conducting this comparison.

1. Task Selection- Two tasks were selected (Wildcard search and select commands) commonly performed by users to retrieve information from MySQL database.
2. User Groups- Five users were selected. Two of which are intensive computer users, two have little knowledge in computer usage and the last user is a Moderate User.
3. Metrics- Metrics for comparison includes:

- Speed- Measures the time taken to complete each task using both voice and keyboard input.
- Accuracy- Compare the accuracy of results achieved through both input methods.
- Error Rate- Record the frequency of errors made with each input method.

4. Experimental Setup- To carry out with this experiment, a time keeper, a laptop and a reliable internet was put in place. Users interacted with the database through the easy-to-use interface and the metrics mentioned in step 3 above calculated and recorded.

5. Data Collection- The predefined tasks using both voice and keyboard inputs were recorded and the result tabulated in tables.

6. The collected results are analyzed and meaningful conclusions drawn.

CHAPTER FOUR

RESULTS AND ANALYSIS

4.1 Presentation of Results

4.1.1 Wildcard Command Method

1st User		Task: Voice Interaction with **MySQL** Database by calling both category and product names		
Age:10years				Computer Usage: Little Knowledge.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	15	0	1
Laptops	HP Omen	16	1	2
Books	Elective Maths	22	1	2
Detergents	Boom Big Size	17	0	1
Detergents	Magic Powder	14	0	1
Tomatoes	Salsa Small	10	0	1
Pen	Ball Pen	13	0	1
Pen	Red Pen	13	0	1
		Total Time=120	Total Error=2	Total Attempts=10

Table 2: User 1 voice Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 1 Voice Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (6 / 8) * 100

Accuracy (%) = 75%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (2 / 10) * 100

Error Rate (%) = 20%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time =120 / 8

Average Time = 15seconds

1st User		Task: Keyboard Interaction with MySQL Database by typing the category and selecting a product name.		
Age:10years				Computer Usage: Little Knowledge.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	6	0	1
Laptops	HP Omen	10	0	1
Books	Elective Maths	6.9	0	1
Detergents	Boom Big Size	5.6	0	1
Detergents	Magic Powder	4.5	0	1
Tomatoes	Salsa Small	3.9	0	1
Pen	Ball Pen	4.7	0	1
Pen	Red Pen	4.7	0	1
		Total Time=46.3	Total Error=0	Total Attempts=8

Table 3:User 1 Keyboard Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 1 Keyboard Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (8 / 8) * 100

Accuracy (%) = 100%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (0 / 8) * 100

Error Rate (%) = 0%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 46.3 / 8

Average Time = 5.8seconds

2nd User		Task: Voice Interaction with MySQL Database by calling both category and product names		
Age:35years				Computer Usage: Little Knowledge.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	13	0	1
Laptops	HP Omen	11	0	1
Books	Elective Maths	11.1	0	1
Detergents	Boom Big Size	11.9	1	2
Detergents	Magic Powder	10.9	0	1
Tomatoes	Salsa Small	11.3	0	1
Pen	Ball Pen	20	1	2
Pen	Red Pen	10	0	1
		Total Time=88.3	Total Error=2	Total Attempts=10

Table 4:User 2 voice Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 2 Voice Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (6 / 8) * 100

Accuracy (%) = 75%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (2 / 10) * 100

Error Rate (%) = 20%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 88.3 / 8

Average Time = 11seconds

2nd User		Task: Keyboard Interaction with MySQL Database by typing the category and selecting a product name.		
Age:36years				Computer Usage: Little Knowledge.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	4.5	0	1
Laptops	HP Omen	5.2	0	1
Books	Elective maths	4.2	0	1
Detergents	Boom Big Size	4.6	0	1
Detergents	Magic Powder	4.5	0	1
Tomatoes	Salsa Small	6.2	0	1
Pen	Ball Pen	4.7	0	1
Pen	Red Pen	5.5	0	1
		Total Time=39.4	Total Error=0	Total Attempts=8

Table 5: User 2 Keyboard Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 2 Keyboard Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (8 / 8) * 100

Accuracy (%) = 100%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (0 / 8) * 100

Error Rate (%) = 0%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 39.4 / 8

Average Time = 4.9seconds

3rd User		Task: Voice Interaction with MySql Database by calling both category and product names		
Age:36 years				Computer Usage: Advance User.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	9	0	1
Laptops	HP Omen	10	1	2
Books	Elective Maths	11	1	2
Detergents	Boom Big Size	10.2	0	1
Detergents	Magic Powder	12.5	0	1
Tomatoes	Salsa small	12	1	3
Pen	Ball Pen	9.5	0	1
Pen	Red Pen	10.2	0	1
		Total Time=84.4	Total Error=3	Total Attempts=12

Table 6: User 3 voice Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 3 Voice Interactions

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (5 / 8) * 100

Accuracy (%) = 62.5%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (3 / 12) * 100

Error Rate (%) = 25%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 84.4 / 8

Average Time = 10.6seconds

3nd User		Task: Keyboard Interaction with MySQL Database by typing the category and selecting a product name.		
Age:36 years				Computer Usage: Advance User.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	4	0	1
Laptops	HP Omen	2.6	0	1
Books	Elective Maths	3	0	1
Detergents	Boom Big Size	2.3	0	1
Detergents	Magic Powder	3.6	0	1
Tomatoes	Salsa small	3	0	1
Pen	Ball Pen	3.4	0	1
Pen	Red Pen	2.5	0	1
		Total Time=24.4	Total Error=0	Total Attempts=8

Table 7:User 3 Keyboard Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 3 Keyboard Interactions

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (8 / 8) * 100

Accuracy (%) = 100%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (0 / 8) * 100

Error Rate (%) = 0%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 24.4 / 8

Average Time = 3.1seconds

4nd User		Task: Voice Interaction with MySQL Database by calling both category and product names		
Age:40 years				Computer Usage: Advance User.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	11	0	1
Laptops	HP Omen	17	1	2
Books	Elective Maths	4	0	1
Detergents	Boom Big Size	30	2	3
Detergents	Magic Powder	13	0	1
Tomatoes	Salsa small	14	0	1
Pen	Ball Pen	20	0	1
Pen	Red Pen	28	2	3
		Total Time=137	Total Error=5	Total Attempts=13

Table 8: User 4 voice Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 4 Voice Interactions

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (5 / 8) * 100

Accuracy (%) = 62.5%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (5 / 8) * 100

Error Rate (%) = 38.5%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 137 / 8

Average Time = 17.1seconds

4nd User		Task: Keyboard Interaction with MySQL Database by typing the category and selecting a product name.		
Age: 40 years				Computer Usage: Advance User.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	5	0	1
Laptops	HP Omen	3.9	0	1
Books	Elective Maths	2.9	0	1
Detergents	Boom Big Size	4.1	0	1
Detergents	Magic Powder	2.8	0	1
Tomatoes	Salsa small	4	0	1
Pen	Ball Pen	2.7	0	1
Pen	Red Pen	3	0	1
		Total Time=28.4	Total Error=0	Total Attempts=8

Table 9: User 4 Keyboard Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 4 Keyboard Interactions

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (8 / 8) * 100

Accuracy (%) = 100%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (0 / 8) * 100

Error Rate (%) = 0%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 28.4 / 8

Average Time = 3.6seconds

5nd User		Task: Voice Interaction with MySQL Database by calling both category and product names		
Age:18years				Computer Usage: Moderate User.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	13	0	1
Laptops	HP Omen	12.7	0	1
Books	Elective Maths	13	0	1
Detergents	Boom Big Size	13	2	3
Detergents	Magic Powder	14	0	1
Tomatoes	Salsa Small	14	2	3
Pen	Ball Pen	13	0	1
Pen	Red Pen	14	0	1
		Total Time=106.7	Total Error=4	Total Attempts=12

Table 10:User 5 Voice Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 5 Voice Interactions

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (4 / 8) * 100

Accuracy (%) = 50%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (4 / 12) * 100

Error Rate (%) = 33.3%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 106.7 / 8

Average Time = 13.3seconds

5nd User		Task: Keyboard Interaction with MySQL Database by typing the category and selecting a product name.		
Age:10years				Computer Usage: .
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	3.9	0	1
Laptops	HP Omen	3.8	0	1
Books	Elective Maths	5.7	0	1
Detergents	Boom Big Size	5.3	0	1
Detergents	Magic Powder	6	1	2
Tomatoes	Salsa Small	4	0	1
Pen	Ball Pen	5.5	0	1
Pen	Red Pen	4	1	2
		Total Time=38.2	Total Error=2	Total Attempts=10

Table 11:User 5 Keyboard Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 5 Keyboard Interactions

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (6 / 8) * 100

Accuracy (%) = 75%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (2 / 10) * 100

Error Rate (%) = 20%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 38.2 / 8

Average Time = 4.8seconds

4.1.2 Select Command Method

1st User		Task: Voice Interaction with MySql Database by calling both category and product names		
Age:10years				Computer Usage: Little Knowledge.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	8	1	2
Laptops	HP Omen	5	0	1
Books	Elective Maths	5	0	1
Detergents	Boom Big Size	7	1	2
Detergents	Magic Powder	5	0	1
Tomatoes	Salsa Small	4.5	0	1
Pen	Ball Pen	4.5	0	1
Pen	Red Pen	5	0	1
		Total Time=44	Total Error=2	Total Attempts=10

Table 12:User 1 Voice Select Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 1 Voice Select Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (6 / 8) * 100

Accuracy (%) = 75%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (2 / 10) * 100

Error Rate (%) = 20%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 44 / 8

Average Time = 5.5seconds

1st User		Task: Keyboard Interaction with MySql Database by typing the category and selecting a product name.		
Age:10years				Computer Usage: Little Knowledge.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	13	1	2
Laptops	HP Omen	14	1	2
Books	Elective Maths	15	0	1
Detergents	Boom Big Size	16	1	2
Detergents	Magic Powder	10	0	1
Tomatoes	Salsa Small	11	0	1
Pen	Ball Pen	9	0	1
Pen	Red Pen	8	0	1
		Total Time=96	Total Error=3	Total Attempts=11

Table 13:User 1 Keyboard Select Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 1 Keyboard Select Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (5 / 8) * 100

Accuracy (%) = 62.5%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (3 / 11) * 100

Error Rate (%) = 27.3%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 96 / 8

Average Time = 12seconds

2nd User		Task: Voice Interaction with MySQL Database by calling both category and product names		
Age:39years				Computer Usage: Little Knowledge.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	5	0	1
Laptops	HP Omen	4	0	1
Books	Elective Maths	4	0	1
Detergents	Boom Big Size	8	1	2
Detergents	Magic Powder	4.5	0	1
Tomatoes	Salsa Small	4.4	1	2
Pen	Ball Pen	5	0	1
Pen	Red Pen	4	0	1
		Total Time=38.9	Total Error=2	Total Attempts=10

Table 4.13 User 2 Voice Select Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 2 Voice Select Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (6 / 8) * 100

Accuracy (%) = 75%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (2 / 10) * 100

Error Rate (%) = 20%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 38.9 / 8

Average Time = 4.9seconds

2nd User		Task: Keyboard Interaction with MySQL Database by typing the category and selecting a product name.		
Age:39years				Computer Usage: Little Knowledge.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	10	0	1
Laptops	HP Omen	12	0	1
Books	Elective maths	13	1	2
Detergents	Boom Big Size	12	0	1
Detergents	Magic Powder	10	0	1
Tomatoes	Salsa Small	10	1	2
Pen	Ball Pen	7	0	1
Pen	Red Pen	6	0	1
		Total Time=80	Total Error=2	Total Attempts=10

Table 14:User 2 Keyboard Select Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 2 Keyboard Select Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (6 / 8) * 100

Accuracy (%) = 75%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (2 / 10) * 100

Error Rate (%) = 20%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 80 / 8

Average Time = 10seconds

3rd User		Task: Voice Interaction with MySQL Database by calling both category and product names		
Age:36 years				Computer Usage: Advance User.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	4	0	1
Laptops	HP Omen	3.5	0	1
Books	Elective Maths	4	0	1
Detergents	Boom Big Size	8	1	2
Detergents	Magic Powder	3.5	0	1
Tomatoes	Salsa small	4.5	0	1
Pen	Ball Pen	4	0	1
Pen	Red Pen	4	0	1
		Total Time=35.5	Total Error=1	Total Attempts=9

Table 4.15 User 3 Voice Select Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 3 Voice Select Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (7 / 8) * 100

Accuracy (%) = 87.5%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (1 / 9) * 100

Error Rate (%) = 11%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 35.5 / 8

Average Time = 4.4seconds

3nd User		Task: Keyboard Interaction with MySql Database by typing the category and selecting a product name.		
Age:36 years				Computer Usage: Advance User.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	7	0	1
Laptops	HP Omen	6	0	1
Books	Elective Maths	6	0	1
Detergents	Boom Big Size	8	0	1
Detergents	Magic Powder	6	0	1
Tomatoes	Salsa small	6	0	1
Pen	Ball Pen	5	0	1
Pen	Red Pen	5.5	0	1
		Total Time=49.5	Total Error=0	Total Attempts=8

Table 15:User 3 Keyboard Select Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 3 Keyboard Select Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (8/ 8) * 100

Accuracy (%) = 100%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (0 / 10) * 100

Error Rate (%) = 0%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 49.5 / 8

Average Time = 6.2seconds

4nd User		Task: Voice Interaction with MySQL Database by calling both category and product names		
Age:40 years				Computer Usage: Advance User.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	4.4	0	1
Laptops	HP Omen	4	0	1
Books	Elective Maths	3.5	0	1
Detergents	Boom Big Size	4	0	1
Detergents	Magic Powder	4.2	0	1
Tomatoes	Salsa small	4	0	1
Pen	Ball Pen	3.5	0	1
Pen	Red Pen	3.6	0	1
		Total Time=31.2	Total Error=0	Total Attempts=8

Table 16: User 4 Voice Select Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 4 Voice Select Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (8 / 8) * 100

Accuracy (%) = 100%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (0 / 10) * 100

Error Rate (%) = 0%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 31.2 / 8

Average Time = 3.9seconds

4nd User		Task: Keyboard Interaction with MySQL Database by typing the category and selecting a product name.		
Age: 40 years				Computer Usage: Advance User.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	10	0	1
Laptops	HP Omen	6	0	1
Books	Elective Maths	9	0	1
Detergents	Boom Big Size	11	0	1
Detergents	Magic Powder	6	0	1
Tomatoes	Salsa small	7	0	1
Pen	Ball Pen	5	0	1
Pen	Red Pen	6	0	1
		Total Time=60	Total Error=0	Total Attempts=8

Table 4.18 User 4 Keyboard Select Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 4 Voice Select Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (8 / 8) * 100

Accuracy (%) = 100%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (0 / 8) * 100

Error Rate (%) = 0%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 60 / 8

Average Time = 7.5seconds

5nd User		Task: Voice Interaction with MySQL Database by calling both category and product names		
Age:18years				Computer Usage: Moderate User.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	4.4	0	1
Laptops	HP Omen	4.5	0	1
Books	Elective Maths	4.5	0	1
Detergents	Boom Big Size	4.5	0	1
Detergents	Magic Powder	4.1	0	1
Tomatoes	Salsa Small	4.2	0	1
Pen	Ball Pen	3.1	0	1
Pen	Red Pen	3.5	0	1
		Total Time=32.8	Total Error=0	Total Attempts=8

Table 17:User 5 Voice Select Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 5 Voice Select Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (8 / 8) * 100

Accuracy (%) = 100%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (0 / 10) * 100

Error Rate (%) = 0%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 32.8 / 8

Average Time = 4.1seconds

5nd User		Task: Keyboard Interaction with MySQL Database by typing the category and selecting a product name.		
Age:18years				Computer Usage: Moderate User.
Category	Product	Time (s)	Error	Number of attempts
Laptops	Dell Vostro	12.1	0	1
Laptops	HP Omen	6.8	2	3
Books	Elective Maths	12.3	0	1
Detergents	Boom Big Size	16.5	0	1
Detergents	Magic Powder	10.4	0	1
Tomatoes	Salsa Small	7.5	0	1
Pen	Ball Pen	4.6	0	1
Pen	Red Pen	6.5	0	1
		Total Time=76.7	Total Error=2	Total Attempts=10

Table 18:User 5 Keyboard Select Interaction

Calculation based on Accuracy, Error Rate and Time taken on User 5 Keyboard Select Interaction

Accuracy (%) = (Number of Correct Results / Total Number of Results) * 100

Accuracy (%) = (6 / 8) * 100

Accuracy (%) = 75%

Error Rate (%) = (Total Number of Errors / Total Number of Attempts) * 100

Error Rate (%) = (2 / 10) * 100

Error Rate (%) = 20%

Average Time =Sum of Time Taken for all Tasks /Total number of tasks

Average Time = 76.7 / 8

Average Time = 9.6seconds

4.1.3 Metrics on Wildcard Search Method

METRICS 1

PERCENTAGE ACCURACY ON BOTH VOICE AND KEYBOARD INPUTS

The Table 19 below displays the result of the wildcard search performed by the five users. The Percentage Accuracy of the Voice Input task against the Keyboard Input task are recorded in the table below.

			VOICE INPUT	KEYBOARD INPUT
Users	Computer Usage	Age of Users	Accuracy (%)	Accuracy (%)
1	Little Knowledge	10yr	75	100
2	Little Knowledge	39yr	70	100
3	Advance User	36yr	62.5	100
4	Advance User	40yr	62.5	100
5	Moderate User	18yr	50	75
			Mean(μ)=64	Mean(μ)=95

Table 19: Wildcard Search Results of the two Tasks on percentage Accuracy

METRICS 2

PERCENTAGE ERROR RATE ON BOTH VOICE AND KEYBOARD INPUTS

The Table 20 below displays the result of the wildcard search performed by the five users. The Percentage Error Rate of the Voice Input task against the Keyboard Input task are recorded in the table below.

			VOICE INPUT	KEYBOARD INPUT
Users	Computer Usage	Age of Users	Error Rate (%)	Error Rate (%)
1	Little Knowledge	10yr	20	0
2	Little Knowledge	39yr	20	0
3	Advance User	36yr	25	0
4	Advance User	40yr	38.5	0
5	Moderate User	18yr	33.3	20
			Mean(μ)=27.36	Mean(μ)=4

Table 20: Wildcard Search Results of the two Tasks on percentage Error rate

METRICS 3

THE AVERAGE TIME TAKEN ON BOTH VOICE AND KEYBOARD INPUTS

The Table 21 below displays the result of the wildcard search performed by the five users. The Average Time taken to complete the Voice Input task against the Keyboard Input task are recorded in the table below.

			VOICE INPUT	KEYBOARD INPUT
Users	Computer Usage	Age of Users	Average Time	Average Time
1	Little Knowledge	10yr	15	5.8
2	Little Knowledge	39yr	11	4.9
3	Advance User	36yr	10.6	3.1
4	Advance User	40yr	17.1	3.6
5	Moderate User	18yr	13.3	4.8
			Mean(μ)=13.4	Mean(μ)=4.44

Table 21:Wildcard Search Results of the two Tasks on Average Time Taken

4.1.4 Metrics on Select Command Method

METRICS 1

The Table 22 below displays the result of the Select Command performed by the five users. The Percentage Accuracy of the Voice Input task against the Keyboard Input task are recorded in the table below.

			VOICE INPUT	KEYBOARD INPUT
Users	Computer Usage	Age of Users	Accuracy (%)	Accuracy (%)
1	Little Knowledge	10yr	75	62.5
2	Little Knowledge	39yr	75	75
3	Advance User	36yr	87.5	100
4	Advance User	40yr	100	100
5	Moderate User	18yr	100	75
			Mean(μ)=87.5	Mean(μ)=82.5

Table 22: Select Command Results of the two Tasks on percentage Accuracy

METRICS 2

PERCENTAGE ERROR RATE ON BOTH VOICE AND KEYBOARD INPUTS

The Table 23 below displays the result of the Select Command performed by the five users. The Percentage Error Rate of the Voice Input task against the Keyboard Input task are recorded in the table below.

			VOICE INPUT	KEYBOARD INPUT
Users	Computer Usage	Age of Users	Error Rate (%)	Error Rate (%)
1	Little Knowledge	10yr	20	27.3
2	Little Knowledge	39yr	20	20
3	Advance User	36yr	11	0
4	Advance User	40yr	0	0
5	Moderate User	18yr	0	20
			Mean(μ)=10.2	Mean(μ)=13.46

Table 23: Select Command Results of the two Tasks on Percentage Error Rate

METRICS 3

Percentage Error Rate on both Voice and Keyboard Inputs

The table 4.26 below displays the result of the Select Command performed by the five users. The Average Time taken to complete the Voice Input task against the Keyboard Input task are recorded in the table below.

			VOICE INPUT	KEYBOARD INPUT
Users	Computer Usage	Age of Users	Average Time	Average Time
1	Little Knowledge	10yr	5.5	12
2	Little Knowledge	39yr	4.9	10
3	Advance User	36yr	4.4	6.2
4	Advance User	40yr	3.9	7.5
5	Moderate User	18yr	4.1	9.6
			Mean(μ)=4.56	Mean(μ)=9.06

Table 24: Select Command Results of the two Tasks on Average Time Taken

4.2 Analysis of Results

The results from the two methods are explained in the following subheadings.

4.2.1 Wildcard Search Method

From Table 19, it is clear that the keyboard input performs better than the voice input. The average Accuracy for Voice Inputs percentage of all five(5) users is 64% compared to 95% for Keyboard Inputs.

The Average Error Rate of Voice Input is also 27.4% compared to 4% for Keyboard input for all five(5) users as seen on Table 20.

From Table 21, it can also be seen that, the average Time taken to complete a Voice Task is 13.4 seconds as compared to 4.4 seconds of the keyboard input.

Figure 16 below shows the average output result for the Wildcard Search method.

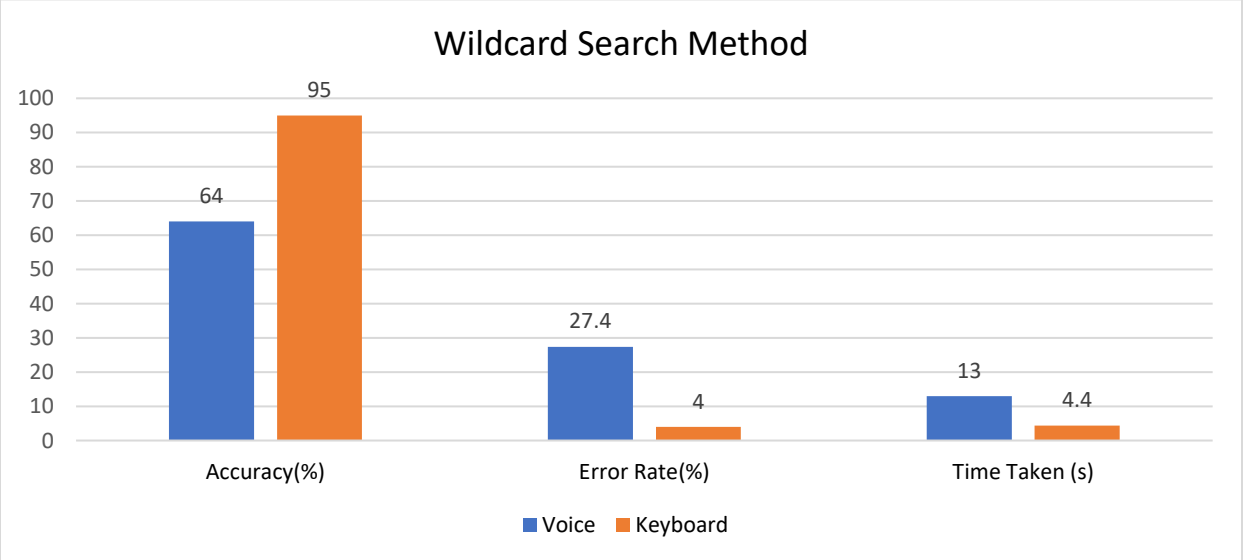


Figure 16: Wildcard Search Result

4.2.2 Select Command Method

From Table 22, it is evident that the voice input performs better than the keyboard input. The average Accuracy for Voice Inputs percentage of all five(5) users is 87.5% compared to 82.5% for Keyboard Inputs.

The Average Error Rate of Voice Input is also 10.2% compared to 13.5% for Keyboard input for all five(5) users as shown on Table 23.

From Table 24, the Average Time taken to complete a Voice Task is 4.6 seconds compared to 9.1 seconds for the keyboard input.

Figure 17 below shows the average output result for the Select Command Method.

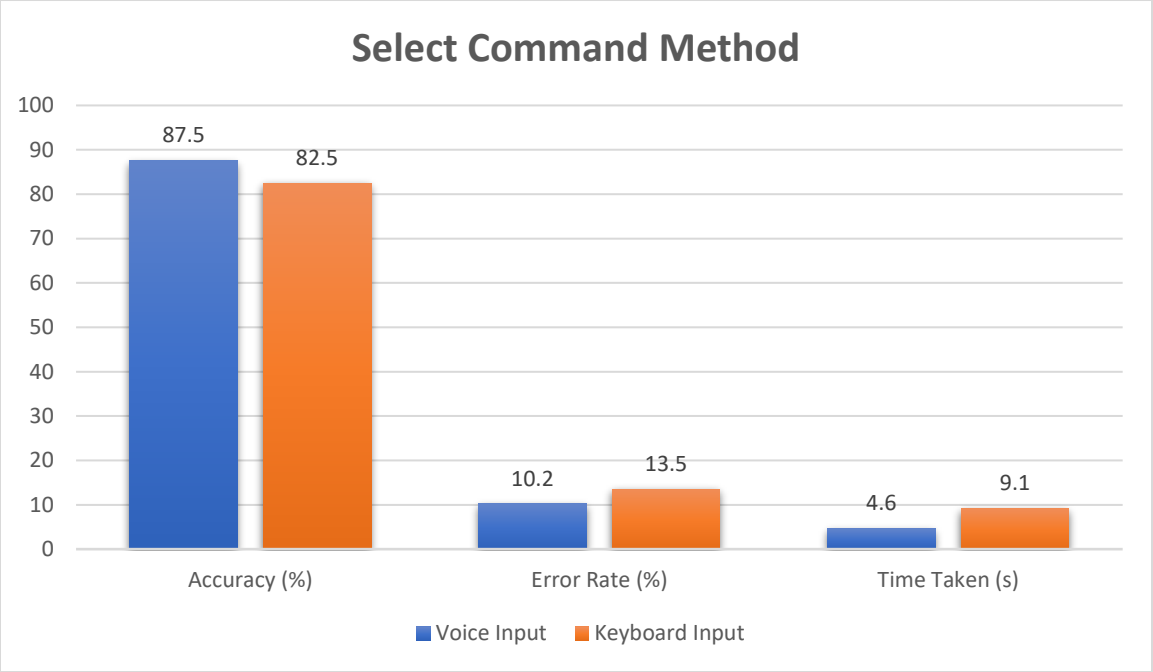


Figure 17: Select Command Resul

CHAPTER FIVE

SUMMARY OF FINDINGS, CONCLUSION AND RECOMMENDATIONS

5.1 Summary of Findings

Regarding SELECT commands, voice input outperformed keyboard input with an average completion time of 4.6 seconds, error rate of 10.2%, and accuracy of 87.5%. In comparison, keyboard input had an average completion time of 9.1 seconds, error rate of 13.5%, and accuracy of 82.5%.

However, voice input faced difficulties when dealing with WILDCARD statements, resulting in a higher error rate of 27.4% compared to 4% for keyboard input. Users often had to repeat their input for clarification, which caused frustration. The average completion time for voice input was 13.4 seconds, with an accuracy of 64%, while keyboard input had an average completion time of 4.4 seconds and an accuracy of 95%.

5.2 Conclusion

The voice input had difficulty with the WILDCARD COMMAND compared to the keyboard input. This could be attributed to the fact that the data used wasn't trained and the diverse range of users interacting with the system. The jargon recognition was particularly challenging, causing frustration for users who had to repeat themselves several times. However, the voice input performed better than the keyboard input regarding the SELECT COMMAND, although it could have still done better with proper training data. Other factors, such as environmental noise and low internet speed, also contributed to the voice input's struggles. Despite all these challenges, a voice command-based interaction with the database has been designed and implemented.

5.3 Recommendations

I suggest adding voice input for SELECT commands, especially for users who need rapid access to information. When it comes to WILDCARD statements, a combination of keyboard and voice input is the preferred option. However, improving voice input for handling wildcards in future updates will make the process completely hands-free. Investigating methods to improve the accuracy and robustness of voice recognition systems, especially in noisy environments or for users with diverse accents and speech patterns.

REFERENCES

Adorf, J. (2013). *Web Speech API*.

agdy Bayoumi, M. A. (1994). *Sphinx: A High-Level Synthesis System for ASIC Design*.

Androutsopoulos, I., Ritchie, G. D., & Thanisch, P. (1995). *Natural Language Interfaces to Databases-An Introduction **.

Barker, J., Marxer, R., Vincent, E., & Watanabe, S. (n.d.). *The third "CHiME" Speech Separation and Recognition Challenge: Dataset, task and baselines*. https://hal.inria.fr/hal-01211376

Burger, J. F. (n.d.). *SEMANTIC DATABASE MAPPING IN EUFID **.

Cristoforetti, L., Ravanelli, M., Omologo, M., Sosi, A., Abad, A., Hagmüller, M., Hagmüller, H., & Maragos, P. (n.d.). *The DIRHA simulated corpus*. http://dirha.fbk.eu

D. Rybach, S. Hahn, P. Lehnen, D. Nolden, M. Sundermeyer, Z. T¨uske, S. Wiesler, R. Schl¨uter, & H. Ney. (n.d.). *RASR – The RWTH Aachen University Open-Source Speech Recognition Toolkit*.

Deshpande, A. K., & Devale, P. R. (2012). NATURAL LANGUAGE QUERY PROCESSING USING PROBABILISTIC CONTEXT FREE GRAMMAR. In *International Journal of Advances in Engineering & Technology* (Vol. 568, Issue 2).

Dooley, K. (n.d.). *Simulation Research Methods*. http://www.eas.asu.edu/~kdooley

Dunn, B., Bandi, K., Kopwa Epse Kassa, C., Shelly, F. U., & Kopwa Epse, C. (2016). *Analysis of the Point of Sales System at Tower Hill Botanical Garden and Suggested Courses of Action* (Vol. 5). https://commons.clarku.edu/sps_masters_papers/5

Fin M, T. de, & González Gutiérrez Dirigido por José Francisco QUESADA, C. (2019). *UNIVERSIDAD DE SEVILLA NATURAL LANGUAGE INTERFACES TO RELATIONAL DATABASES*.

Gaikwad, M. P. (2008). Natural Language Interface to Database. In *Certified International Journal of Engineering and Innovative Technology (IJEIT)* (Vol. 9001, Issue 8).

Giordani, A., & Moschitti, A. (2009). Semantic mapping between natural language questions and SQL queries via syntactic pairing. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5723 LNCS*, 207–221. https://doi.org/10.1007/978-3-642-12550-8_17

Gross Barbara J, Appelt Douglas E, Paul Martin, & Pereira Fernando. (1986). *TEAM: An Experiment in the Design of Transportable Natural-Language Interface*.

Harris, L. R. (1984). *Technology Transfer-Experience with INTELLECT: Artificial Intelligence Technology Transfer*.

Hendrix, G. G., Sacerdoti, E. D., Sagalowicz, D., & Slocum, J. (n.d.). *Developing a Natural Language Interface to Complex Data*.

Henisz Thompson, B., & Thompson, F. B. (n.d.). *ASK Is Transportable in Half a Dozen Ways*.

International Islamic University Malaysia Kulliyyah of Engineering, Annual IEEE Computer Conference, International Conference on Computer and Communication Engineering 2010.05.11-12 Kuala Lumpur, & ICCCE 2010.05.11-12 Kuala Lumpur. (n.d.). *International Conference on Computer and Communication Engineering (ICCCE), 2010 11-12 May 2010, Kuala Lumpur, Malaysia*.

Joshi, M., & Srivastava, S. R. (2015). Human Computer Interaction Using Speech Recognition Technology. *International Bulletin of Mathematical Research*, *2*(1).

Kabir, M. A., & Han, B. (2016). An improved usability evaluation model for point-of-sale systems. *International Journal of Smart Home*, *10*(7), 269–282. https://doi.org/10.14257/ijsh.2016.10.7.27

Kim, K. G. (2016). Book Review: Deep Learning. *Healthcare Informatics Research*, *22*(4), 351. https://doi.org/10.4258/hir.2016.22.4.351

Lee, A., & Kawahara, T. (n.d.). *Recent Development of Open-Source Speech Recognition Engine Julius*. http://julius.sourceforge.jp

Li, F., & Jagadish, H. v. (2014). *Constructing an Interactive Natural Language Interface for Relational Databases *.*

Linarès, G., Nocera, P., Massonié, D., & Matrouf, D. (2007). The LIA speech recognition system: From 10×RT to 1×RT. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4629 LNAI*, 302–308. https://doi.org/10.1007/978-3-540-74628-7_40

Mahajan, R., Roy, A., Jadhav, A., Rao, A., Gosavi, H., & Patil, S. (n.d.). *DATABASE INTERACTION USING SPEECH RECOGNITION*. www.ijariie.com432

Mohri, M. (n.d.). *Finite-State Transducers in Language and Speech Processing*.

Naidu, A. (2009). Factors affecting patient satisfaction and healthcare quality. *International Journal of Health Care Quality Assurance*, *22*(4), 366–381. https://doi.org/10.1108/09526860910964834

Nihalani, M. N., Silakari, S., & Motwani, M. (2011). Natural language Interface for Database: A Brief review. *IJCSI International Journal of Computer Science Issues*, *8*(2). www.IJCSI.org

Nixon, R. (n.d.). *Learning PHP, MySQL, and JavaScript*.

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (n.d.). *LIBRISPEECH: AN ASR CORPUS BASED ON PUBLIC DOMAIN AUDIO BOOKS*. http://www.gutenberg.org

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., Facebook, Z. D., Research, A. I., Lin, Z., Desmaison, A., Antiga, L., Srl, O., & Lerer, A. (n.d.). *Automatic differentiation in PyTorch*.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlíček, P., Qian, Y., Schwarz, P., Silovsk´y, J. S., Stemmer, G., & Vesel´y, K. V. (n.d.). *The Kaldi Speech Recognition Toolkit*. http://kaldi.sf.net/

Rao, G., Agarwal, C., Chaudhry, S., Kulkarni, N., & Patil, S. (2010). NATURAL LANGUAGE QUERY PROCESSING USING SEMANTIC GRAMMAR. In *IJCSE) International Journal on Computer Science and Engineering* (Vol. 02, Issue 02).

Ravanelli, M., Brakel, P., Omologo, M., & Bengio, Y. (2017). *A network of deep neural networks for distant speech recognition*. http://arxiv.org/abs/1703.08002

Renals, S., Hain, T., & Bourlard, H. (n.d.). *INTERPRETATION OF MULTIPARTY MEETINGS THE AMI AND AMIDA PROJECTS*. http://www.amiproject.org/

Rosnik Philip. (1989). *ACCESS TO MULTIPLE UNDERLYINGI SYSTEMS IN JANUS*. https://apps.dtic.mil/sti/pdfs/ADA214585.pdf

Saravjeet Kaur, & Rashmeet Singh Bali. (2012). SQL GENERATION AND EXECUTION FROM NATURAL. *International Journal of Computing & Business Research*.

Soto-Acosta, P., Martinez-Conesa, I., & Colomo-Palacios, R. (n.d.). *An Empirical Analysis of the Relationship Between IT Training Sources and IT Value*.

Suehring Steve, & Valade Janet. (n.d.). *PHP, MySQL ®, Javascript ® & HTML5 A L L-I N-O N E*.

Temiz, H. (n.d.). *DeepSR: A Deep Learning Tool for Single Image Super Resolution*. https://doi.org/10.13140/RG.2.2.31239.73125

Templeton, M., & Burger, J. (n.d.). *PROBLEMS IN NATURAL-LANGUAGE INTERFACE TO DSMS WITH EXAMPLES FROM EUFID*.

Thansekhar, M. R., & Balaji, N. (n.d.). *Database Interaction using Automatic Speech Recognition*.

Toosi, K. K., Impink, B. G., Baker, N. A., & Boninger, M. L. (2011). Effects of computer keyboarding on ultrasonographic measures of the median nerve. *American Journal of Industrial Medicine*, *54*(11), 826–833. https://doi.org/10.1002/ajim.20983

w3schools.com. (n.d.). *Browser Statistics*. Retrieved March 3, 2023, from https://www.w3schools.com/browsers/default.asp

Wahi, V., Avadhoot, J., Rohan, P., Mujawar, A., & Tayde, N. (n.d.). *Voice Controlled Database Analysis*. www.ijert.org

Warren, D. H. D., & Pereira, F. C. N. (1982). *An Efficient Easily Adaptable System for Interpreting Natural Language Queries 1*.

Young, S. (1994). *The HTK Hidden Markov Model Toolkit: Design and Philosophy Open Domain Spoken Dialogue Systems View project Face Recognition Ph.D. View project*. https://www.researchgate.net/publication/263124034

Yu, D., & Deng, L. (n.d.). *Signals and Communication Technology Automatic Speech Recognition A Deep Learning Approach*. http://www.springer.com/series/4748

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). LibriSpeech: *An ASR Corpus Based on Public Domain Audio Books.*

Laurenzano, M. A., Clark, J. H., Lee, G. G., Stoler, A., Hemphill, C. L., & Touretzky, D. S. (2020). *Mozilla Common Voice: A Massively Multilingual Speech Corpus.*

Chung, J. S., Nagrani, A., & Zisserman, A. (2019). *The VoxCeleb Speaker Recognition Challenge 2019.*

Sak, H., Senior, A., & Beaufays, F. (2014). *A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition*.

Warden, P. (2018). Speech Commands: *A Dataset for Limited-Vocabulary Speech Recognition.*

Chan, W., Jaitly, N., Le, Q., & Vinyals, O. (2016). Listen, Attend and Spell. In Proceedings of the 2016 IEEE International Conference on *Acoustics, Speech and Signal Processing (ICASSP)*

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., ... & Coates, A. (2014). Deep Speech*: Scaling up end-to-end speech recognition*.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., ... & Coates, A. (2017). Deep Speech 2: *End-to-End Speech Recognition in English and Mandarin*

Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist Temporal Classification: *Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In Proceedings of the 23rd International Conference on Machine Learning (ICML).*

Gulati, A., Gangireddy, D., Alamri, H., Dehak, N., Wang, Y., Povey, D., ... & Saon, G. (2020). Conformer: *Convolution-augmented Transformer for Speech Recognition. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*