

# **MINNI INTEGRATED VOICE ASSSISTANCE**

## **A PROJECT REPORT**

*Submitted by*

**KARTHIKEYAN C – 210701111**

**THOUFEEQ MOHAMMED M – 210701503**

**RAJA SEKAR M - 210701512**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**



**RAJALAKSHMI ENGINEERING COLLEGE**

**ANNA UNIVERSITY, CHENNAI**

**MAY 2024**

# **RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI**

## **BONAFIDE CERTIFICATE**

Certified that this Thesis titled “**MINNI INTEGRATED VOICE ASSISTANCE**” is the bonafide work of “**KARTHIKEYAN C (210701111), THOUFEEQ MOHAMMED (210701503), RAJA SEKAR (210701512)**” who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

### **SIGNATURE**

Dr . K.Anand M.E.,Ph.D.,

### **PROJECT COORDINATOR**

Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on\_\_\_\_\_

**Internal Examiner**

**External Examiner**

## **ABSTRACT**

NIMMI sounds like a truly innovative addition to the world of digital assistants! Its focus on advanced natural language processing and machine learning algorithms, coupled with its platform-agnostic design, sets it apart in a rapidly evolving landscape. The emphasis on privacy and security is also commendable, addressing important concerns that users have with digital assistants. Its versatility in integration across various ecosystems, including smart homes, wearables, and automotive systems, makes it a promising tool for users seeking a seamless and consistent experience across different devices and platforms. Furthermore, the commitment to transparency and user control over their data adds an extra layer of trust and reliability. In a competitive market where new technologies and startups are constantly emerging, NIMMI's focus on personalized responses, enhanced privacy features, and wide integration capabilities could indeed give it a significant edge. As users increasingly rely on voice assistants for various tasks, innovations like NIMMI can enhance the overall user experience and make technology more accessible and user-friendly. It's exciting to see how NIMMI will continue to evolve and unlock new possibilities in how we interact with our digital environments. With ongoing advancements in natural language understanding and integration capabilities, there's certainly a lot of potential for NIMMI to shape the future of voice-activated technology.

## ACKNOWLEDGMENT

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan B.E., F.I.E.**, for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan Ph.D.**, for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan B.E., M.S.**, for providing us with the requisite infrastructure.

We also express our sincere gratitude to our college Principal, **Dr. S. N. Murugesan M.E., PhD.**, and **Dr. P. KUMAR M.E., PhD, Director computing and information science**, and **Head Of Department of Computer Science and Engineering** and our project coordinator **Dr. K.Anand M.E.,Ph.D.**, for her encouragement and guiding us throughout the project towards successful completion of this project and to our parents, friends, all faculty members and supporting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

**KARTHIKEYAN C**  
**THOUFEEQ MOHAMMED M**  
**RAJA SEKAR M**

## **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	<b>iii</b>
	<b>LIST OF TABLES</b>	<b>v</b>
	<b>LIST OF FIGURES</b>	<b>vii</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 PROBLEM STATEMENT	
	1.2 SCOPE OF THE WORK	
	1.3 AIM AND OBJECTIVES OF THE PROJECT	
	1.4 RESOURCES	
	1.5 MOTIVATION	
<b>2.</b>	<b>LITERATURE SURVEY</b>	<b>4</b>

<b>3.</b>	<b>SYSTEM DESIGN</b>	<b>7</b>
	3.1 GENERAL	
	3.2 SYSTEM ARCHITECTURE DIAGRAM	
	3.3 DEVELOPMENT ENVIRONMENT	
	3.3.1 HARDWARE REQUIREMENTS	
	3.3.2 SOFTWARE REQUIREMENTS	
<b>4.</b>	<b>PROJECT DESCRIPTION</b>	<b>9</b>
	4.1 METHODOLOGY	
<b>5.</b>	<b>RESULTS AND DISCUSSIONS</b>	<b>10</b>
	5.1 FINAL OUTPUT	
	5.2 RESULT	
<b>6.</b>	<b>CONCLUSION AND FUTURE</b>	
<b>7.</b>	<b>ENHANCEMENT</b>	
	7.1 CONCLUSION	
	7.2 FUTURE ENHANCEMENT	
	<b>APPENDIX</b>	<b>15</b>
	<b>REFERENCES</b>	<b>30</b>

	<b>LIST OF FIGURES</b>	
<b>FIGURE NO</b>	<b>TITLE</b>	<b>PAGE NO</b>
3.2.1	SYSTEM ARCHITECTURE	7
5.1	OUTPUT	8
5.2	SAD MINNI	11
5.3	HAPPY MINNI	12

# **CHAPTER 1**

## **INTRODUCTION**

In the quickly changing environment of digital assistants, NIMMI stands out as a pioneering integrated voice assistant that is meant to flow smoothly into the fabric of daily life. NIMMI, created with the goal of improving user engagement using powerful natural language processing (NLP) and machine learning algorithms, provides a customized and intuitive user experience, establishing a new standard in the field of voice-activated technology.

At its heart, NIMMI is designed to interpret and process complicated inquiries with surprising precision, utilizing cutting-edge AI to provide solutions that are not just relevant but contextually aware. This functionality means that NIMMI can assist with a wide range of activities, including smart home device management and real-time information, as well as personal productivity and entertainment options. This versatile capability distinguishes NIMMI as a significant instrument in modern life, capable of increasing productivity and enjoyment in a variety of daily activities.

One of NIMMI's distinguishing advantages is its integration flexibility. NIMMI is designed to be platform-agnostic, so it can work across several ecosystems, providing users with a consistent and dependable assistance independent of device or operating system. This integration includes smart homes, wearables, and even automobile systems, giving NIMMI a truly pervasive presence in consumers' life. Because NIMMI can work smoothly across several platforms and devices, users can rely on it for ongoing support whether they are at home, on the move, or in their automobiles.

Furthermore, NIMMI promotes privacy and security, using strong encryption and anonymization techniques to safeguard user information. In an age where data security is critical, NIMMI's developers have pledged openness and control for users, guaranteeing that interactions with NIMMI are secure and private. Users may be certain that their personal information is secure, allowing them to communicate freely with the assistant.

A new voice assistant like NIMMI could be focused on specific functionalities, integration capabilities, or user interfaces to distinguish itself in a competitive market. If NIMMI is a recent development, it could incorporate advanced features beyond those found in earlier voice assistants, such as improved natural language understanding, more personalized responses, enhanced privacy features, or integration with a wider range of devices and services. Startups and established tech companies alike are continually seeking ways to innovate in this space, making voice assistants more useful, intuitive, and integrated into our daily lives.

NIMMI is leading the way in the development of the upcoming voice assistant generation, providing unmatched security, usability, and integration. It is a promising addition to the digital assistant market, with the potential to completely change the way we engage with technology in our daily lives thanks to its superior skills and dedication to privacy.



## **1.1 PROBLEM STATEMENT**

In the fast-changing world of digital assistants, we really need one that can blend seamlessly into our daily lives while also offering advanced features and top-notch security. A lot of the current voice assistants struggle with handling complex questions, giving relevant responses, and working smoothly across different platforms and systems. On top of that, users are rightly worried about privacy and keeping their data safe. NIMMI steps in to tackle these problems by using cutting-edge natural language processing and machine learning to provide accurate and contextually relevant solutions. It's designed to work consistently across smart homes, wearables, and car systems. Plus, NIMMI makes user privacy a top priority by using strong encryption and anonymization techniques to ensure secure interactions. Despite all these cool features, the challenge is to keep innovating and improving NIMMI to stay ahead in a competitive market and fully unleash its potential to change how we interact with everyday technology.

## **1.2 SCOPE OF THE WORK**

NIMMI's work covers a wide range of functions and integration capabilities, all aimed at seamlessly fitting into your daily life. It uses advanced NLP and AI to understand and respond to complex questions, making various tasks more efficient and enjoyable. NIMMI is designed to work with different ecosystems like smart homes, wearables, and car systems, ensuring consistent support across different devices. To prioritize privacy and security, NIMMI uses strong encryption and anonymization to safeguard user data. As a cutting-edge voice assistant, it sets a new standard for user interaction, providing personalized and intuitive experiences. NIMMI is always evolving to include advanced features such as better understanding of natural language, more personalized responses, and integration with a wider range of devices, solidifying its position as a game-changer in the digital assistant market.

## **1.3 AIM AND OBJECTIVES OF THE PROJECT**

NIMMI is all about changing the game for digital assistants. It seamlessly fits into your daily routine, using advanced natural language processing (NLP) and machine learning to keep you engaged. The goal is to give you a personalized and intuitive experience that sets a new standard for voice-activated tech. NIMMI can handle complex questions with precision and offers context-aware solutions for managing your smart home, getting real-time info, boosting your productivity, and keeping you entertained. It works smoothly with different ecosystems like smart homes, wearables, and cars, making sure you get consistent support across the board. And when it comes to privacy and security, NIMMI's got your back with strong encryption and anonymization to keep your data safe. With top-notch features like improved NLP, personalized responses, and seamless device integration, NIMMI stands out in a crowded market. This cutting-edge voice assistant is all about transforming how you interact with tech, making it more accessible, fun, and a crucial part of your everyday life.

## **1.4 RESOURCES**

In the fast-changing world of digital assistants, NIMMI stands out as a cutting-edge voice assistant that's designed to seamlessly fit into your daily life. By using advanced natural language processing (NLP) and machine learning, NIMMI aims to give you a personalized and user-friendly experience, raising the bar for voice-activated technology. It's really good at understanding and dealing with complex questions, thanks to its top-notch AI that provides contextually aware solutions. NIMMI can help with a wide range of tasks, like managing smart home devices, giving you real-time info, and boosting your productivity and entertainment. And it's designed to work well across different platforms, including smart homes, wearables, and automotive systems. NIMMI takes your privacy and security seriously, using strong encryption and anonymization methods to keep your data safe, and it's all about being transparent and giving you control. With its advanced features like improved NLP, personalized responses, and extensive device integration, NIMMI really stands out in a crowded market. It's all set to shake up the voice assistant industry by offering unbeatable usability, security, and integration, changing the way we use technology in our everyday lives.

## **1.5 MOTIVATION**

NIMMI is all about staying ahead in the world of digital assistants. It's all about making your everyday life easier and more connected. NIMMI uses super smart language skills and learning from experience to set a whole new standard for voice-activated tech. It's all about giving you a personalized and easy experience. NIMMI can handle even the trickiest questions with total accuracy, giving you solutions that fit exactly what you need, making everything you do more efficient and fun. NIMMI is designed to work smoothly on any platform, whether it's your smart home, wearable tech, or in your car. And, of course, keeping your privacy and security in mind, NIMMI keeps everything super safe with top-notch encryption and privacy tools. With its advanced features and commitment to being cutting edge, NIMMI is leading the way for the next generation of voice assistants, changing the way we use technology every day.

## **CHAPTER 2**

### **LITRETURE SURVEY**

In [1] Current systems that use Automatic Speech Recognition (ASR) technology to extract data from the BIM model are unable to automate the entire process, retrieve a broad variety of data, or enable remote engagement. This is especially problematic for users who are disabled. The paper fills this theoretical and methodological gap by providing a two-way, automated, and agnostic solution. To demonstrate the applicability, a "Proof of Concept" prototype was created using Amazon Alexa, an AI voice assistant platform. The result demonstrates the validity of the information that was created and retrieved. The suggested solution's elements, such as the AI voice assistant interface and mediation environment to translate spoken requests and retrieve data to CSV files, also have a high degree of compatibility. Subsequent investigations endeavor to expand this resolution to retrieve data from a BIM cloud model, augmenting its usability and accessibility even more.

In [2] This study proposes a comprehensive research model that extends the technological acceptance model (TAM) to encompass usability and emotional needs elements in older individuals' adoption of voice assistants (VAs), addressing the urgent worldwide issue of population aging. The study emphasizes the significance of emotional needs being met and emphasizes the vital function of companionship in affecting the intention to adopt VA use through empirical validation with 425 older users. Remarkably, there was no significant correlation identified between behavioral intention and perceived ease of use, indicating that ease of use might not be the only factor driving adoption. This research offers developers and policymakers in the fields of assistive technologies and geriatric care useful information for creating virtual assistants (VAs) customized for the senior population. This study calls for more user-centered approaches in future VA development and advances a deeper understanding of how VAs might improve older individuals' well-being by giving emotional demands priority in VA design.

In [3] This research focuses on using wireless sensor networks to create an Internet of Things Android voice assistant that will improve higher education. Convenient intelligent learning and management capabilities are made possible by the system's integration of sensor data collecting, transmission, and analysis with the school network platform. The findings show that the Android voice assistant efficiently improves learning and teaching efficiency by enabling voice commands for course inquiries, resource procurement, teaching evaluations, and other duties. The system delivers intelligent learning and management features for students, professors, and administrators through the synergy of sensor data collecting and wireless transmission capabilities, promising breakthroughs in the higher education network environment.

In [4] This long-term field study explores how well-informed kids are about voice assistants (VAs), with an emphasis on parental interventions and kids' communicative behaviors—like spilling secrets—when interacting with VAs. Children have a limited knowledge of data storage, but they can infer information about how VAs process data based on the qualitative findings. Parental interventions improve children's comprehension of data processing, but not storage, indicating a need for systems that can explain themselves. Furthermore, children's readiness to divulge secrets to virtual assistants is

adversely correlated with their awareness of data storage, underscoring the significance of a foundational grasp of technology for younger users.

In [5] This study looks into the effects of voice-based personal assistants' (VPAs) various intelligence qualities on user adoption and confidence. It detects competence and integrity trust in VPAs along with functional, hedonic, and anthropomorphic traits by drawing on social agency theory and the technology acceptance model. The findings show that whereas anthropomorphic qualities strengthen integrity, functional and hedonic features foster competence. Integrity and competence both have a good impact on positive word-of-mouth, and competency also has an impact on usage intention. Furthermore, the growth of trust in VPAs is moderated by the user's degree of technology apprehension. This study sheds light on consumer trust in VPAs from a theoretical and practical perspective.

In [6] In order to close the knowledge gap regarding their application for public, private, and active transportation, this study looks at the function of ICTs and new digital technologies in transportation communication. It analyzes 158 papers and uses a methodical literature assessment to identify interesting digital technologies for study and implementation. The results emphasize the need for information collection and dissemination to transportation users, with a particular focus on the function of mobile applications in delivering tailored trip planning and real-time updates. The report highlights implementation gaps in digital literacy and desire to use, but also points out that many technologies are still in the trial or research stages and have the potential to improve the accessibility of transportation-related information.

In [7] The goal of this project is to use mobile sensor networks to improve the voice recognition capabilities of intelligent learning support systems for English education. Learners' voice signals are obtained from various sensor nodes, preprocessed, and feature-extracted for enhanced recognition accuracy by using a speech recognition algorithm based on these networks. Comparing experimental findings to older approaches, they show higher performance in speaker change and background noise settings. This technology provides more accurate and dependable voice recognition for learners, which enhances learning outcomes and interactive experiences and makes it a more useful learning tool for English language learners.

In [8] This study addresses the difficulties that children with hearing impairments encounter while using conventional teaching techniques by introducing a mobile application game. It talks about the goals, research topics, and how to use cutting-edge mobile technologies to support effective learning. A design model for the creation of the mobile application game is presented, and the literature study looks at the learning challenges faced by children with hearing impairments. The design of the voice recognition mobile app is also shown in a block diagram. The program uses tactics from behavior modification theories to improve learning motivation, interest in the material, self-efficacy, and attitudes toward studying in people with hearing impairments.

In [9] The author offers a customized Android-based mobile EFL learning platform that is geared on vocabulary acquisition. The system, which is made up of a Data Synchronization Intelligence Agent, Client Mobile Learning System, and Remote Management Server, adjusts to the needs and memory stages of each individual learner. The system's efficacy in improving English learning performance was confirmed by the large improvements in vocabulary capacity indicated by evaluation through pre- and post-exam multiple-choice and cloze tests. The usefulness and applicability of the method in enhancing learners' language abilities is further validated by the simulation.

In [10] In the process of developing mobile apps, this work presents the Out of Step structural method for cross-language clone detection. The approach uses enriched concrete syntax trees to find commonalities between various programming languages, hence addressing the problem of synchronizing code bases in polyglot systems. Evaluation shows that Kotlin and Dart have over 80% similarity in language features and can effectively recognize clones for sorting algorithms with a precision of 67% to 95%. Furthermore, the algorithm's capability to detect code similarities throughout the full logic of an application is validated using a corpus of 144 mobile apps written in Kotlin and Dart, providing a promising foundation to lessen the effects of app synchronization in multi-language environments.

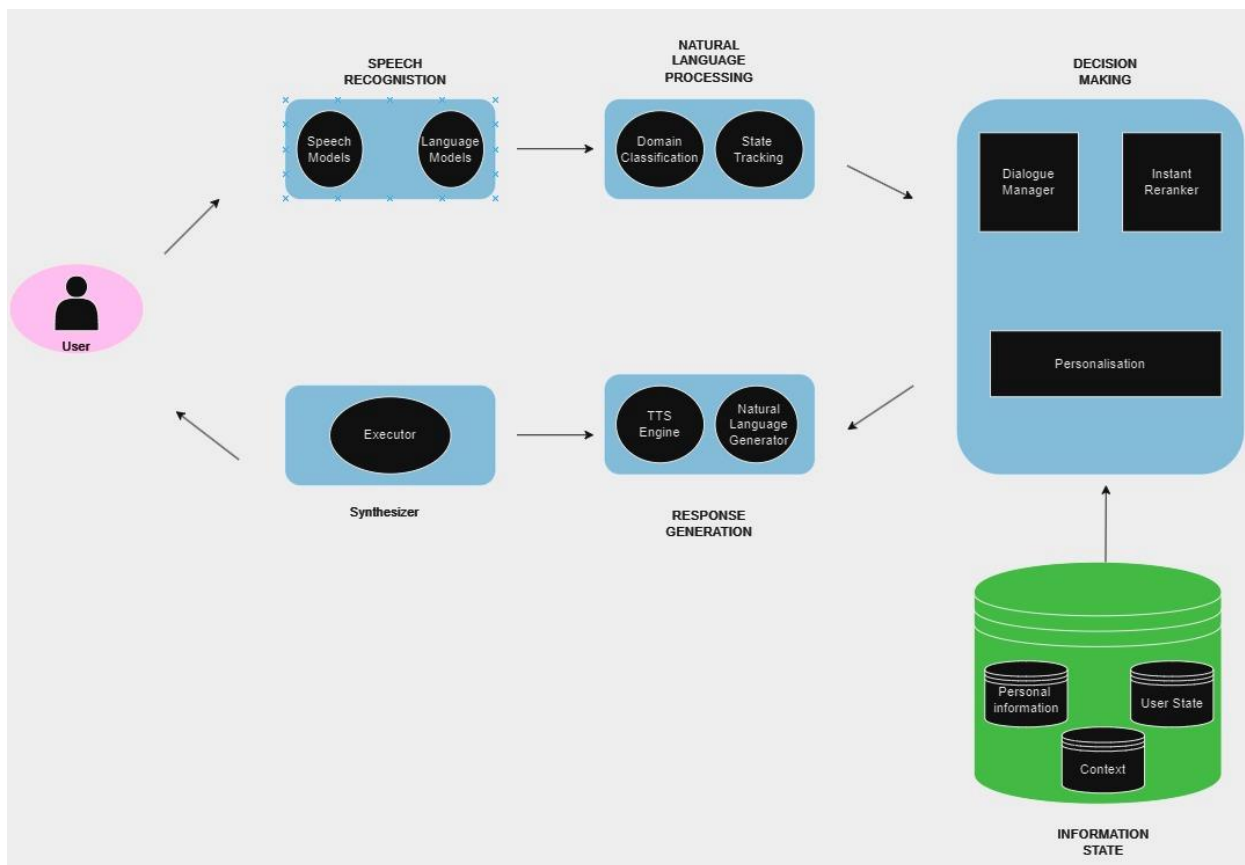
## CHAPTER 3

### SYSTEM DESIGN

#### 3.1 GENERAL

In this section, we would like to show how the general outline of how all the components end up working when organized and arranged together. It is further represented in the form of a flow chart below.

#### 3.2 SYSTEM ARCHITECTURE DIAGRAM



**Fig 3.2.1 SYSTEM ARCHITECTURE DIAGRAM**

## 3.1 DEVELOPMENTAL ENVIRONMENT

### 3.1.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the system's implementation. It should therefore be a complete and consistent specification of the entire system. It is generally used by software engineers as the starting point for the system design.

**Table 3.1 Hardware Requirements**

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i3
RAM	8 GB RAM
MONITOR	15" COLOR
HARD DISK	512 GB
PROCESSOR SPEED	MINIMUM 1.1 GHz

### 3.1.2 SOFTWARE REQUIREMENTS

The software requirements document is the specifications of the system. It should include both a definition and a specification of requirements. It is a set of what the system should rather be doing than focus on how it should be done. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating the cost, planning team activities, performing tasks, tracking the team, and tracking the team's progress throughout the development activity.

**Visual Studio** and **chrome** would all be required.

## **CHAPTER 4**

### **PROJECT DESCRIPTION**

#### **4.1 METHODOLOGY**

The methodology outlined in the provided text involves the development and implementation of NIMMI, an integrated voice assistant. It incorporates powerful natural language processing (NLP) and machine learning algorithms to improve user engagement. The process involves interpreting and processing complex inquiries with precision, utilizing cutting-edge AI for relevant and contextually aware solutions. NIMMI's versatility allows it to assist with various activities, including smart home device management, real-time information retrieval, personal productivity, and entertainment options. Its platform-agnostic design ensures seamless integration across multiple ecosystems, including smart homes, wearables, and automobile systems. Additionally, NIMMI prioritizes privacy and security, employing strong encryption and anonymization techniques to safeguard user information, thus ensuring secure and private interactions. The methodology underscores a commitment to innovation, aiming to incorporate advanced features and integration capabilities to distinguish NIMMI in the competitive voice assistant market and enhance its utility and integration into users' daily lives.

The methodology outlined in the text revolves around the development and implementation of NIMMI as a groundbreaking integrated voice assistant. It highlights the use of advanced natural language processing (NLP) and machine learning algorithms to enhance user engagement and provide a personalized user experience. The approach emphasizes NIMMI's ability to interpret and process complex queries with precision, leveraging AI to deliver contextually aware responses. Furthermore, the methodology emphasizes NIMMI's integration flexibility, designed to operate seamlessly across various ecosystems and devices. Privacy and security are paramount, with NIMMI employing robust encryption and anonymization techniques to protect user data. Overall, the methodology focuses on creating a voice assistant that not only understands human language nuances but also anticipates user needs and preferences, thereby enhancing the digital experience.

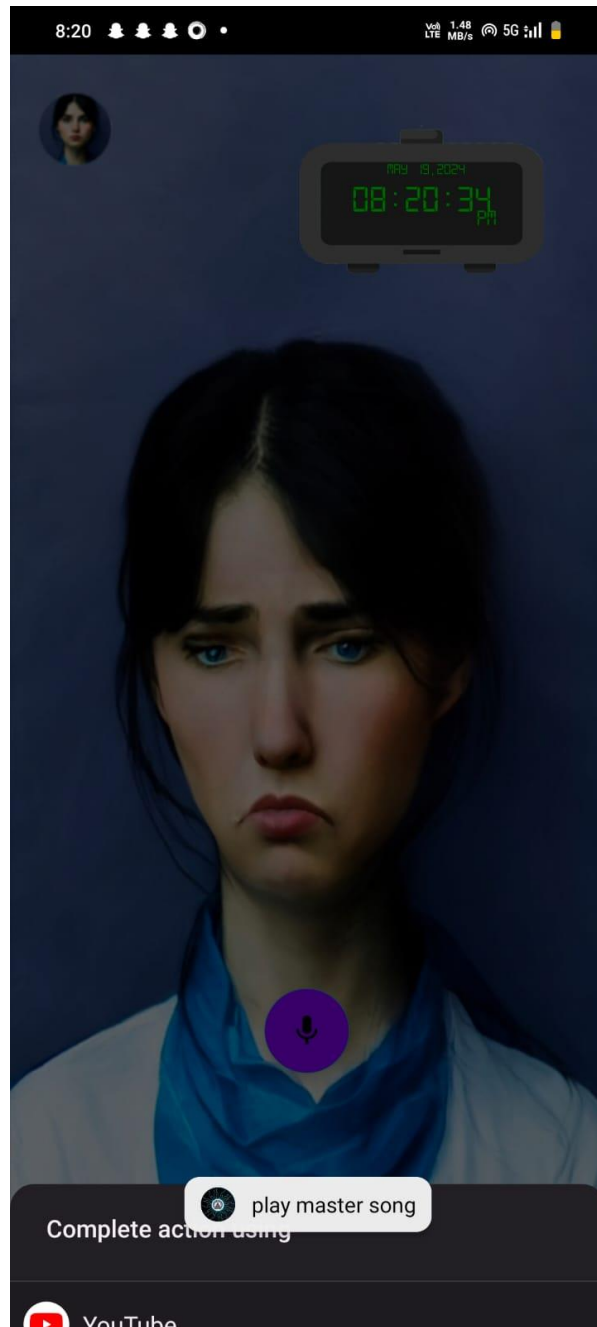


## CHAPTER 5

### RESULTS AND DISCUSSIONS

#### 5.1 OUTPUT

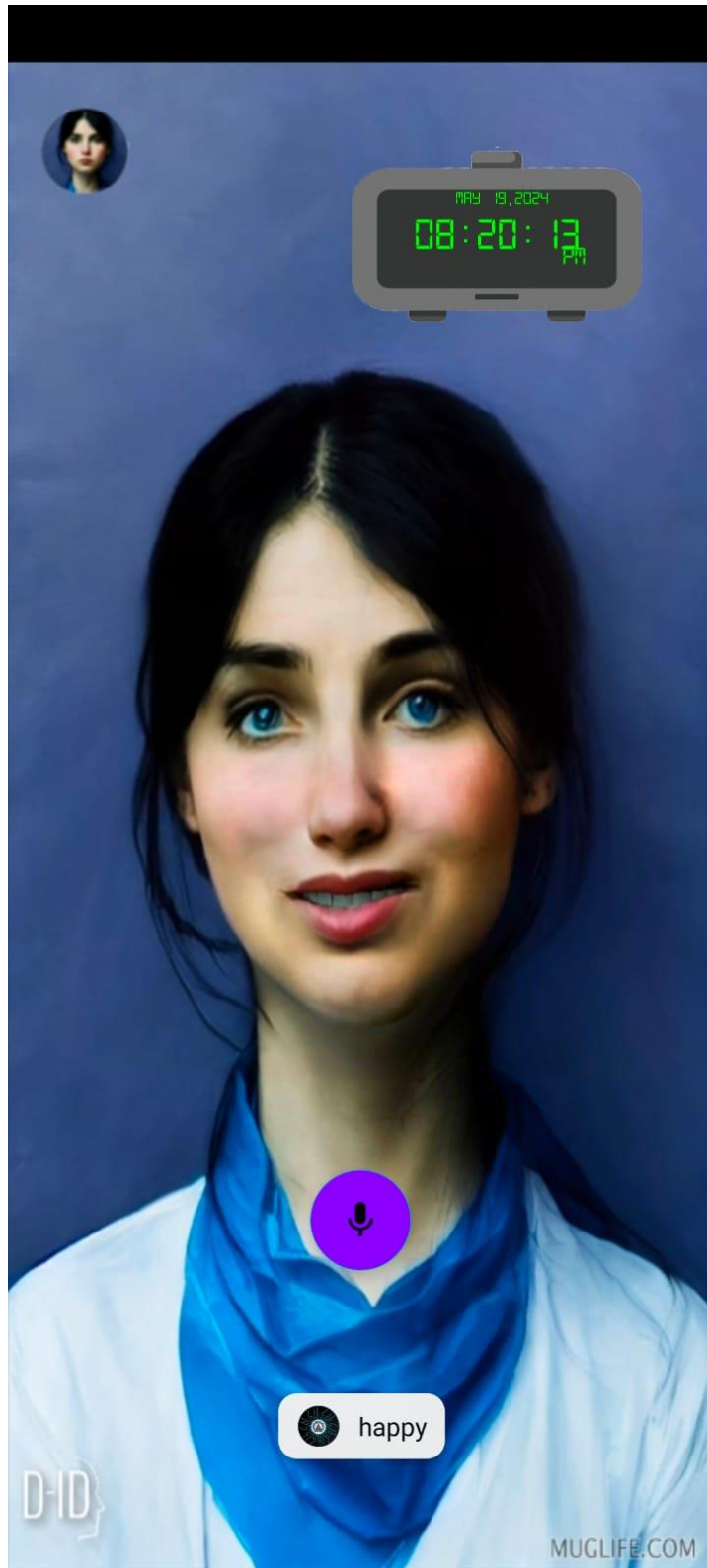
The following images contain images attached below of the working application.



**Fig 5.1: Output**



**Fig 5.2: Sad Minni**



**Fig 5.3: Happy Minni**

## **5.2 RESULT**

NIMMI, as a pioneering integrated voice assistant, aims to seamlessly integrate into daily life, offering a customized user experience powered by advanced NLP and machine learning algorithms. Its ability to interpret complex queries with precision enables assistance in various tasks, from managing smart home devices to enhancing personal productivity and entertainment options. NIMMI's integration flexibility across multiple platforms ensures consistent support, while prioritizing privacy and security through robust encryption and anonymization techniques fosters user trust. Positioned as a leader in the voice assistant market, NIMMI sets new standards with its focus on security, usability, and integration, promising to revolutionize daily technology interactions with its advanced features and commitment to privacy.

## **CHAPTER 6**

### **CONCLUSION AND FUTURE ENHANCEMENT**

#### **6.1 CONCLUSION**

In conclusion, NIMMI represents a significant advancement in the realm of voice-activated technology, offering unparalleled integration flexibility, privacy, and security. Its cutting-edge AI, precision in interpreting inquiries, and seamless operation across various platforms make it a pivotal tool in modern life. With a focus on user engagement, privacy, and personalized experiences, NIMMI sets a new standard for voice assistants. Its ability to adapt to different ecosystems and devices ensures consistent and reliable assistance for users, while its commitment to privacy and security instills confidence in its usage. As technology continues to evolve, NIMMI's dedication to innovation and user-centric design positions it as a leader in the next generation of voice assistants, poised to reshape our daily interactions with technology in a secure and seamless manner.

#### **6.2 FUTURE ENHANCEMENT**

In Future enhancements for NIMMI could focus on expanding its capabilities even further to meet evolving user needs and technological advancements. This may involve integrating cutting-edge advancements in natural language understanding to enhance its ability to comprehend and respond to complex queries more accurately and contextually. Additionally, further personalization features could be incorporated to tailor responses and interactions to individual user preferences and habits, thereby providing a more immersive and personalized experience. Enhancements in privacy and security could involve the implementation of even more robust encryption and anonymization techniques, along with increased transparency and user control over data usage. Integration with emerging technologies such as augmented reality or virtual reality could also offer new avenues for user interaction and engagement. Moreover, continuous updates and improvements to NIMMI's software and algorithms will be essential to ensure it remains at the forefront of voice assistant technology and continues to set new standards in usability, security, and integration across various platforms and devices.

## APPENDIX

### SOURCE CODE:

```
package com.example.nimmifinal.login;

//import static com.example.login.Functions.fetchName;

import static com.example.nimmifinal.R.id.profile_nimmi;
import static com.example.nimmifinal.R.id.rec_but;
import static com.example.nimmifinal.login.functions.wishMe;

import android.annotation.SuppressLint;
import android.app.SearchManager;
import android.content.Context;
import android.content.Intent;
import android.graphics.Typeface;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.provider.MediaStore;
import android.provider.Settings;
import android.speech.RecognitionListener;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;
import android.speech.tts.TextToSpeech;
import android.text.format.DateUtils;
import android.view.View;
import android.view.WindowManager;
import android.widget.TextClock;
import android.widget.TextView;
import android.widget.Toast;

import android.widget.VideoView;
import androidx.annotation.NonNull;
import androidx.annotation.RequiresPermission;
import androidx.appcompat.app.AppCompatActivity;

import com.example.nimmifinal.R;
import com.example.nimmifinal.login.MainActivity;
import com.example.nimmifinal.login.MainActivity3;
import com.firebase.client.Firebase;
import com.google.android.material.floatingactionbutton.FloatingActionButton;

import java.util.ArrayList;
import java.util.Date;

import kotlinx.coroutines.BuildersKt;
import android.Manifest;
```

```

import android.content.pm.PackageManager;
import android.os.Bundle;
import android.speech.RecognitionListener;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import java.util.ArrayList;
import java.util.Locale;

public class MainActivity2 extends AppCompatActivity {
    VideoView videoView;

    private SpeechRecognizer recognizer;
    private TextView tvResult;
    private TextToSpeech tts;
    private BuildersKt Dexter;
    static int PERMISSION_CODE= 100;
    FloatingActionButton callbtn;
    Firebase firebase;
    private static final int REQUEST_RECORD_AUDIO_PERMISSION = 200;
    private SpeechRecognizer speechRecognizer;

    @SuppressWarnings("MissingInflatedId")
    @Override
    public void onCreate(Bundle savedInstanceState) {

        TextClock textClock, textampm, textdate;
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
        textClock = findViewById(R.id.clock);
        textampm = findViewById(R.id.ampm);
        textdate = findViewById(R.id.date);
        callbtn = findViewById(R.id.callbtn);
        Typeface myfornt = Typeface.createFromAsset(getApplicationContext().getAssets(),
"font/digital_fornt.ttf");
        textClock.setTypeface(myfornt);
        textdate.setTypeface(myfornt);
        textampm.setTypeface(myfornt);

        firebase.setAndroidContext(this);
        String                               uniqueId                               =

```

```

Settings.Secure.getString(getApplicationContext().getContentResolver(),
Settings.Secure.ANDROID_ID);
    firebase = new Firebase("https://nimmi-1d18d-default-rtdb.firebaseio.com/User" +
uniqueId);
    FloatingActionButton rec = (FloatingActionButton) findViewById(rec_but);

    tvResult = findViewById(R.id.tv_result);
    tvResult.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            final String name = tvResult.getText().toString();

            Firebase child_name = firebase.child("Speak");
            child_name.setValue(name);
            if (name.isEmpty()) {
                tvResult.setError("this is an required field");
                tvResult.setEnabled(false);
            } else {
                tvResult.setError(null);
                tvResult.setEnabled(true);
            }
        }
    });

    // Dexter.withContext(this)
    //     .withPermission(Manifest.permission.RECORD_AUDIO)
    //     .withListener(new PermissionListener() {
    //         @Override public void onPermissionGranted(PermissionGrantedResponse response)
    /* ... */
    //         @Override public void onPermissionDenied(PermissionDeniedResponse response)
    /* ... */
    //         @Override public void onPermissionRationaleShouldBeShown(PermissionRequest
permission, PermissionToken token) { /* ... */
    //     }).check();
    //     findById();
    initializeTextToSpeech();
    initializeResult();
    //     fetchName("call Amma and say hello to her");

    videoView = findViewById(R.id.videoview);
    Uri uri = Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.nimmi_bk);
    videoView.setVideoURI(uri);

    videoView.start();

    videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

```



```

        @Override
        public void onPrepared(MediaPlayer mp) {
            // mp.setLooping(true);
        }
    });

    TextView txtButton = findViewById(profile_nimmi);
    // imgButton.setSize(imgButton.SIZE_STANDARD);
    txtButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            openpro();
        }

    });

    setContentView(R.layout.activity_main2);

    // Check for microphone permission
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (checkSelfPermission(Manifest.permission.RECORD_AUDIO) !=
        PackageManager.PERMISSION_GRANTED) {
            requestPermissions(new String[]{Manifest.permission.RECORD_AUDIO},
            REQUEST_RECORD_AUDIO_PERMISSION);
        }
    }

    // Initialize SpeechRecognizer
    speechRecognizer = SpeechRecognizer.createSpeechRecognizer(this);
    speechRecognizer.setRecognitionListener(new RecognitionListener() {
        @Override
        public void onReadyForSpeech(Bundle params) {
        }

        @Override
        public void onBeginningOfSpeech() {
        }

        @Override
        public void onRmsChanged(float rmsdB) {
        }

        @Override
        public void onBufferReceived(byte[] buffer) {
        }

        @Override
        public void onEndOfSpeech() {

```

```

    }

    @Override
    public void onError(int error) {
    }

    @Override
    public void onResults(Bundle results) {
        // Process the recognized speech
        ArrayList<String> matches =
results.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
        if (matches != null) {
            for (String result : matches) {
                if (result.toLowerCase(Locale.getDefault()).contains("wake up")) {
                    // Detected wake-up word, trigger action
                    Toast.makeText(MainActivity2.this, "Wake-up word detected!",
Toast.LENGTH_SHORT).show();
                    response("hi");
                    {
                        Intent intent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
                        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
                        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE,
Locale.getDefault());
                        speechRecognizer.startListening(intent);
                    }
                }
            }
        }
    }

    @Override
    public void onPartialResults(Bundle partialResults) {
    }

    @Override
    public void onEvent(int eventType, Bundle params) {
    }
});

// Start listening for speech

}

//@Override

```

```

private void openpro() {

    Intent intent = new Intent(this, MainActivity9.class);
    startActivity(intent);
}

@Override
protected void onResume() {
//    videoView.resume();
    super.onResume();
}

@Override
protected void onRestart() {
    videoView.start();
    super.onRestart();
}

@Override
protected void onPause() {
    videoView.suspend();
    super.onPause();
}

@Override
protected void onDestroy() {
    videoView.stopPlayback();
    super.onDestroy();
}
}

public void openActivity3() {
    Intent intent = new Intent(this, MainActivity3.class);
    startActivity(intent);
}

public void openActivity5() {
    Intent intent = new Intent(this, MainActivity.class);
    startActivity(intent);
}

}

private void initializeTextToSpeech() {
    tts=new TextToSpeech(this, new TextToSpeech.OnInitListener() {
        @Override

```

```

        public void onInit(int i) {
            if (tts.getEngines().size()==0){

                Toast.makeText(MainActivity2.this, "Engine is not available ",
Toast.LENGTH_SHORT).show();
                }else{ String s = wishMe();

                    //speak(s);
                }
            }
        });
    }

    public void name(){
        videoView=findViewById(R.id.videoview);
        Uri i= Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.nim_name);
        videoView.setVideoURI(i);
        videoView.start();

        videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
            @Override
            public void onPrepared(MediaPlayer mp) {

                // mp.setLooping(true);
            }
        });

    }

    public void angry() {
        videoView = findViewById(R.id.videoview);
        Uri i = Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.angry);
        videoView.setVideoURI(i);
        videoView.start();

        videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
            @Override
            public void onPrepared(MediaPlayer mp) {

                // mp.setLooping(true);
            }
        });
    }

    public void sad(){
        videoView=findViewById(R.id.videoview);

```

```

Uri i= Uri.parse("android.resource://" +getPackageName()+ "/" +R.raw.sad);
videoView.setVideoURI(i);
videoView.start();

videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
    @Override
    public void onPrepared(MediaPlayer mp) {

        // mp.setLooping(true);
    }
});

}

public void loop(){
    videoView=findViewById(R.id.videoview);
    Uri i= Uri.parse("android.resource://" +getPackageName()+ "/" +R.raw.nimmi_bk);
    videoView.setVideoURI(i);
    videoView.start();

    videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
        @Override
        public void onPrepared(MediaPlayer mp) {

            // mp.setLooping(true);
        }
    });
}

public void hello(){
    videoView=findViewById(R.id.videoview);
    Uri i= Uri.parse("android.resource://" +getPackageName()+ "/" +R.raw.nim_hello);
    videoView.setVideoURI(i);
    videoView.start();

    videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
        @Override
        public void onPrepared(MediaPlayer mp) {

            // mp.setLooping(true);
        }
    });
}

public void lust(){
    videoView=findViewById(R.id.videoview);

```

```

Uri i= Uri.parse("android.resource://" +getPackageName()+ "/" +R.raw.lre);
videoView.setVideoURI(i);
videoView.start();

videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
    @Override
    public void onPrepared(MediaPlayer mp) {

        // mp.setLooping(true);
    }
});

}

public void happy(){
    videoView=findViewById(R.id.videoview);
    Uri i= Uri.parse("android.resource://" +getPackageName()+ "/" +R.raw.happy);
    videoView.setVideoURI(i);
    videoView.start();

    videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
        @Override
        public void onPrepared(MediaPlayer mp) {

            // mp.setLooping(true);
        }
    });
}

private void speak(String msg) {
    tts.speak(msg, TextToSpeech.QUEUE_FLUSH, null, null);
}

private void initializeResult() {
    if (SpeechRecognizer.isRecognitionAvailable(this)) {
        recognizer = SpeechRecognizer.createSpeechRecognizer(this);
        recognizer.setRecognitionListener(new RecognitionListener() {
            @Override
            public void onReadyForSpeech(Bundle bundle) {

            }

            @Override
            public void onBeginningOfSpeech() {

```

```

    }

    @Override
    public void onRmsChanged(float v) {

    }

    @Override
    public void onBufferReceived(byte[] bytes) {

    }

    @Override
    public void onEndOfSpeech() {

    }

    tvResult = findViewById(R.id.tv_result);
    tvResult.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View v) {

            final String name=tvResult.getText().toString();

            Firebase child_name=firebase.child("Speak");
            child_name.setValue(name);
            if (name.isEmpty()){
                // tv_result.setError("this is an required field");
                tvResult.setEnabled(false);
            }else
            {
                // tv_result.setError(null);
                tvResult.setEnabled(true);
            }
        }
    });

    }

    @Override
    public void onError(int i) {
        try{
            Thread.sleep(200);
        } catch (InterruptedException e) {
            speak("oop sorry, i can't understand.. but you may give a report to server");
            e.printStackTrace();
        }
        //openActivity3();
        // speak("oop sorry, i can't understand.. but you may give a report to server");
    }

```

```

    }

    @Override
    public void onResults(Bundle bundle) {
        ArrayList<String> result =
bundle.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
        Toast.makeText(MainActivity2.this, "" + result.get(0),
Toast.LENGTH_SHORT).show();
        // tvResult.setText(result.get(0));
        response(result.get(0));
    }
    @Override
    public void onPartialResults(Bundle bundle) {

    }

    @Override
    public void onEvent(int i, Bundle bundle) {

    }
    });
}
}
public void response(@NonNull String msg) {

```

```

String msgs = msg.toLowerCase();

```

```

if (msgs.contains("call")) {
    openActivity5();
}

else if (msgs.startsWith("hi", 0)) {
    // speak("Hello nice to see you");
    hello();
} else if (msgs.contains("hello")) {
    //speak(" Hello , Tell me how can I help you");
    hello();

} else if (msgs.matches("sad||sad face||face of sad")) {
    //speak(" Hello , Tell me how can I help you");
    sad();

}
else if (msgs.matches("lust||lust face||")) {
    //speak(" Hello , Tell me how can I help you");

```



```

        lust();

    }

    else if (msgs.matches("anger||angry||angry face||face of angry")) {
        //speak(" Hello , Tell me how can I help you");
        angry();
    }
    else if (msgs.matches("happy||happy||happy face||smiling face")) {
        //speak(" Hello , Tell me how can I help you");
        happy();
    } else if (msgs.endsWith("your name")) {
        // speak("Okay, let me introduce myself I am NEMMI, An artificial intelligence, i can
do many things when compare to other AI");
        name();

    } // else if (msgs.contains("nimmi")) {
        // speak("yeah ,tell me ");
        //hello();}
    else if (msgs.contains("how are you")) {
        speak("i am doing great,what about you");

    } else if (msgs.endsWith("report")) {
        speak(" you can send to the server");
        openActivity3();
    } else if (msgs.endsWith("time")) {

        Date date = new Date();
        String time = DateUtils.formatDateTime(this, date.getTime(),
DateUtils.FORMAT_SHOW_TIME);
        speak(("The time is" + time));

    } else if (msgs.endsWith("open arlo")) {
        speak(" you can send to the server");
        // openActivity6();
    } else if (msgs.endsWith("date")) {

        Date date = new Date();
        String todays_date = DateUtils.formatDateTime(this, date.getDate(),
DateUtils.FORMAT_SHOW_DATE);
        speak(("Todays date is" + todays_date));

    }

    if (msgs.endsWith("google")) {
        speak("opening google chrome");
        Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.google.com"));
        startActivity(intent);
    }

```

```

        else if (msgs.contains("open")) {
            if (msgs.contains("browser")) {
                speak("opening google chrome");
                Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.google.com"));
                startActivity(intent);
            }
            if (msgs.endsWith("chrome")) {
                speak("opening google chrome");
                Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.google.com"));
                startActivity(intent);
            }
            if (msgs.endsWith("open youtube")) {
                speak("opening YouTube");
                Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.youtube.com"));
                startActivity(intent);
            }
            } else if (msgs.contains("open")) {
                if (msgs.contains("facebook")) {
                    speak("opening Facebook");
                    Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.facebook.com"));
                    startActivity(intent);
                }
            } else if (msgs.endsWith("song")) {
                //Intent intent=msgs.replace()

                // Uri uri = Uri.parse("http://www.youtube.com/search?q=" + msgs);
                Intent intent = new
Intent(MediaStore.INTENT_ACTION_MEDIA_PLAY_FROM_SEARCH);

                intent.putExtra(MediaStore.EXTRA_MEDIA_FOCUS,
MediaStore.Audio.Playlists.ENTRY_CONTENT_TYPE);
                intent.putExtra(SearchManager.QUERY,msgs);// The user's search query
                intent.putExtra(MediaStore.EXTRA_MEDIA_TITLE,msgs); // Set the title of the
media item you want to play

                intent.putExtra(MediaStore.EXTRA_MEDIA_FOCUS,
MediaStore.Audio.Playlists.ENTRY_CONTENT_TYPE);
                intent.putExtra(MediaStore.EXTRA_MEDIA_TITLE,msgs); // Set the title of the
media item you want to play
                intent.putExtra(SearchManager.QUERY,msgs); // The user's search query
                startActivity(intent);
            }
        }
    }
}

```

```

    } else if (msgs.startsWith("tell me ", 0)) {
        Intent intent = new Intent(Intent.ACTION_WEB_SEARCH);
        intent.putExtra (SearchManager.QUERY, msgs);
        startActivity(intent);
    }
    if (msg.endsWith("photos")) {
        speak("opening albums");
        Intent i = new Intent(MediaStore.ACTION_PICK_IMAGES);
        startActivity(i);
    }
    if (msg.endsWith("album")) {
        speak("opening albums");
        Intent i = new Intent(MediaStore.ACTION_PICK_IMAGES);
        startActivity(i);
    }
    if (msg.endsWith("camera")) {
        speak("opening camara");
        Intent i = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivity(i);
    }
    } else if (msgs.contains("record")) {
        if (msg.contains("video")) {
            speak("recording video");
            Intent i = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
            startActivity(i);
        }
    }
    if (msgs.endsWith("spotify")) {
        speak("opening spotify");
        Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("https://open.spotify.com/search/"+msgs));
        startActivity(intent);
        // Context ctx = this;
        //Intent intent =
        ctx.getPackageManager().getLaunchIntentForPackage("com.spotify.android");
        //startActivity(intent);

    } else if (msgs.endsWith("youtube")){
        Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.youtube.com/results?search_query="+msgs));
        startActivity(intent);
    }
    else if (msgs.contains("open")) {
        if (msgs.contains("whatsapp")) {
            speak("opening whatsapp");

            Context ctx = this;

```

```

        Intent intent =
ctx.getPackageManager().getLaunchIntentForPackage("com.whatsapp.android");
        startActivity(intent);
    }
    } else if (msgs.contains("open")) {
        if (msgs.contains("twitter")) {
            speak("opening twitter");
            Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.twitter.com"));
            startActivity(intent);
        }
    }

    if (msgs.endsWith("9994 920 257")) {
        Intent i = new Intent(Intent.ACTION_CALL);
        i.setData(Uri.parse(msgs));
        startActivity(i);
    }
    /* else { {
        Intent intent = new Intent(Intent.ACTION_WEB_SEARCH);
        intent.putExtra(SearchManager.QUERY,msgs);
        startActivity(intent);

    }

    /*if (msgs.endsWith("9994 920 257" )) {
        Intent i = new Intent(Intent.ACTION_CALL);
        i.setData(Uri.parse(msgs));
        startActivity(i);} */

    }

    public void StartRecording(View view){

        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,RecognizerIntent.LANGU
AGE_MODEL_FREE_FORM);
        intent.putExtra(RecognizerIntent.EXTRA_MAX_RESULTS,1);
        recognizer.startListening(intent);

    }

}

```

## REFERENCES

1. Elghaish, F., Chauhan, J. K., Matarneh, S., Rahimian, F. P., & Hosseini, M. R. (2022, August 1). Artificial intelligence-based voice assistant for BIM data management. *Automation in Construction*. <https://doi.org/10.1016/j.autcon.2022.104320>
2. Liu, M., Wang, C., & Hu, J. (2023, November 1). Older adults' intention to use voice assistants: Usability and emotional needs. *Heliyon*. <https://doi.org/10.1016/j.heliyon.2023.e21932>
3. Ye, W., & Li, M. (2024, March 1). Application of IoT Android voice assistant based on sensor networks in higher education network mode. *Measurement. Sensors*. <https://doi.org/10.1016/j.measen.2024.101091>
4. Szczuka, J. M., Strathmann, C., Szymczyk, N., Mavrina, L., & Krämer, N. C. (2022, September 1). How do children acquire knowledge about voice assistants? A longitudinal field study on children's knowledge about how voice assistants store and process data. *International Journal of Child-computer Interaction*. <https://doi.org/10.1016/j.ijcci.2022.100460>
5. Huang, R., Kim, M., & Lennon, S. (2024, June 1). Voice-Based Personal Assistant (VPA) Trust: Investigating Competence and Integrity. *Telematics and Informatics Reports*. <https://doi.org/10.1016/j.teler.2024.100140>
6. Yigitcanlar, T., Downie, A. T., Mathews, S., Fatima, S., MacPherson, J., Behara, K. N., & Paz, A. (2024, January 1). Digital technologies of transportation-related communication: Review and the state-of-the-art. *Transportation Research Interdisciplinary Perspectives*. <https://doi.org/10.1016/j.trip.2023.100987>
7. Jingning, L. (2024, April 1). Speech recognition based on mobile sensor networks application in English education intelligent assisted learning system. *Measurement. Sensors*. <https://doi.org/10.1016/j.measen.2024.101084>
8. Yue, W. S., & Zin, N. A. M. (2013, January 1). Voice Recognition and Visualization Mobile Apps Game for Training and Teaching Hearing Handicaps Children. *Procedia Technology*. <https://doi.org/10.1016/j.protcy.2013.12.218>
9. Huang, X. (2023, January 1). Design and Application of English Assisted Learning System Based on Mobile Learning Platform. *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2023.11.027>
10. Jimenez, S., Rakic, G., Takahashi, S., & Cardozo, N. (2024, September 1). *Out of Step: Code Clone Detection for Mobile Apps Across Different Language Codebases*. *Science of Computer Programming*. <https://doi.org/10.1016/j.scico.2024.103112>

