A

Project Report
on

AirSwipe: An AI-Driven Presentation Controller With Gesture and Voice Recognition

Submitted to the Savitribai Phule Pune University in Fulfillment of the Requirement for the Award of BE in Computer Engineering by

Ms. Ayesha Shaikh [B401200229]
Ms. Pooja Gupta [B401200218]
Ms. Shiba Shaikh [B401200230]
Ms. Anushka Jadhav [B401200167]

Under the guidance of

Prof. Pradnya K. Bachhav

Department of Computer Engineering

Guru Gobind Singh College of Engineering and Research Centre

Nashik-422009

2024-25

GURU GOBIND SINGH COLLEGE OF ENGINEERING AND RESEARCH CENTRE

Nashik-422009

2024-2025

Department of Computer Engineering

CERTIFICATE

This is to certify that the PROJECT REPORT entitled

AirSwipe: An AI-Driven Presentation Controller With Gesture and Voice Recognition

is submitted as fulfilment of the

Project Examination BE in Computer Engineering

BY

Ms. Ayesha Shaikh [B401200229]
Ms. Pooja Gupta [B401200218]
Ms. Shiba Shaikh [B401200230]
Ms. Anushka Jadhav [B401200167]

Prof. Pradnya.K Bachhav
Project Guide

Prof. Pradnya. K. Bachhav & Prof. P. C. Patil
Project Coordinator

Prof. S. G. Shukla
Head of the Department

Dr. N. G. Nikam
Principal

SAVITRIBAI PHULE PUNE UNIVERSITY

CERTIFICATE

This is to certify that,

Ms. Ayesha Shaikh [B401200229]

Ms. Pooja Gupta [B401200218]

Ms. Shiba Shaikh [B401200230]

Ms. Anushka Jadhav [B401200167]

of BE in Computer Engineering was examined in the

Project Examination entitled

AirSwipe: An AI-Driven Presentation Controller With Gesture and Voice Recognition

on

__ /06/2025

At

Department of Computer Engineering

GURU GOBIND SINGH COLLEGE OF ENGINEERING AND RESEARCH CENTRE

Nashik-422009

2024-2025

---------- ----------

Internal Examiner *External Examiner*

Acknowledgement

It is a great pleasure to acknowledge those who extended their support, and contributed time for this project work.

As this project has been developed, We would like to thank our project guide **Mrs.Pradnya K. Bachhav**, for her valuable and skillful guidance, assessment and suggestions from time to time improved the quality of work in all respects. We would like to take this opportunity to express my deep sense of gratitude towards her, for her invaluable contribution in completion of this project.

We are also thankful to **Mr. Sandeep. G. Shukla**, Head of Computer Engineering Department for his timely guidance, inspiration and administrative support without which my work would not have been completed.

We are also thankful to the all staff members of Computer Engineering Department and Librarian, Guru Gobind Singh College of Engineering and Research Centre, Nashik.

Also we would like to thank our colleagues and friends who helped us directly and indirectly to complete this project.

Ms. Ayesha Shaikh
Ms. Pooja Gupta
Ms. Shiba Shaikh
Ms. Anushka Jadhav

Abstract

The idea AI Presentation Tool Using Gesture Recognition introduces an innovative AI-driven presentation tool that utilizes hand gestures for controlling PowerPoint slides, aiming to transform the presentation experience. By enabling presenters to interact with their slides through natural movements, the tool eliminates the reliance on traditional physical devices, such as remotes or keyboards, fostering a more fluid and engaging atmosphere. The system employs advanced computer vision and deep learning techniques to accurately recognize hand gestures in real-time, providing a hands-free solution that is particularly beneficial for users with physical disabilities. The proposed tool not only simplifies the act of presenting but also enhances accessibility, making it an ideal fit for diverse environments, including classrooms, corporate meetings, conferences, and virtual settings. By leveraging recent advancements in AI technology, the proposed idea aspires to redefine the standards of interactivity and inclusivity in presentation tools, making presentations more dynamic and accessible to a wider audience. In this paper we have included study of existing papers and methodology of proposed system

Keywords:-*AI, presentation tool, computer vision, classroom teaching, speech recognition, machine learning.*

Abbreviation

Sr No.	Abbriviation	Full Form
1	HCI	Human Computer Interface
2	AI	Artificial Intelligence
3	GUI	Graphical User Interface
4	CNN	Convolutional Neural Network
5	gTTS	Google Text-to-Speech
6	API	Application Programming Interface
7	TTS	Text-to-Speech

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Presentations play a crucial role in both professional and academic communication. Traditional presentation tools, such as clickers and keyboards, often disrupt the natural flow of the presenter, diverting attention from the audience and focusing it on managing the slides. These conventional devices can also create challenges for users with disabilities, limiting interactivity and accessibility for those who may find the tools difficult to use.

To address these issues, we propose an innovative AI-driven presentation tool that leverages gesture recognition technology, allowing for hands-free slide navigation. By utilizing advanced computer vision and deep learning algorithms, this tool can interpret simple hand gestures in real time, enabling the presenter to control the presentation without needing physical controllers. This enhances the presenter's engagement with the audience while creating a more accessible and interactive experience.

The system's intuitive interface facilitates a dynamic presentation environment where the presenter can focus entirely on delivering their message. Additionally, the tool promotes inclusivity by offering an equal platform for all users, including professionals, educators, and individuals with disabilities. The tool could potentially be expanded to support additional features, such as triggering multimedia elements or adjusting settings through specific gestures.

As AI and machine learning continue to revolutionize human-computer interaction, this project represents a step towards more intuitive, hands-free technology, transforming not only how we present but also how we engage with digital content across various domains, from education to entertainment and assistive technology.

1.1 Overview

The AI Presentation Tool is an innovative system that enhances human-computer interaction (HCI) by enabling effortless control of presentations through hand gestures and voice commands. By combining AI-driven computer vision and natural language

processing, the tool can recognize gestures and interpret voice commands in real time. Developed using Python, it incorporates powerful frameworks such as MediaPipe for gesture recognition, OpenCV for image processing, and speech recognition technologies for voice command integration. Additionally, the tool offers accessibility features like live speech-to-text subtitles and translation, making it a more inclusive option for users with varying physical abilities. Its platform-independent design and customizable features ensure that it can meet the unique needs of presenters across educational, professional, and assistive environments.

1.2 Aim

The primary aim of this project is to develop a hands-free, AI-powered presentation tool that allows users to control slides and multimedia content through gesture recognition and voice commands, enhancing user interaction, accessibility, and presentation efficiency.

1.3 Objectives

1. To develop accurate and reliable gesture recognition capabilities for seamless slide control.
2. To integrate voice command functionality for a user-friendly, hands-free experience.
3. To support customizable gestures and voice commands to suit individual user preferences.
4. To implement real-time speech-to-text conversion and translation for enhanced accessibility.
5. To ensure the tool is inclusive and accessible for users with different physical capabilities.
6. To design a responsive and intuitive graphical user interface (GUI).

1.4 Organization of Report

The rest of this report is organized in the following manner. In all chapters, related contents are described in detail.

- **Introduction (Chapter 1)**: In this chapter, the overview of existing systems and their problem is discussed. This chapter describes the aim, motivation, and objectives of the software system.

- **Literature Survey (Chapter 2)**: In this chapter, Related work done in the Previous papers have advantages and disadvantages. Related information is available in standard Books, Journals, Transactions, Internet Websites, etc. is discussed.
- **Software Requirement Specification (Chapter 3)**: In this chapter, the detailed description of requirements is specified.
- **System Design (Chapter 4)**: This chapter discusses the proposed system with the help of system architecture, system design, and UML diagrams
- **Technical Specifications (Chapter 5)**: This chapter, discusses the technical details used in the project
- **Project Estimation Schedule and Team Structure (Chapter 6)**: This chapter discusses project estimate, brief of COCOMO model, and related calculation and team structure
- **Software Implementation (Chapter 7)**: This chapter discusses important module and algorithm also business logic and archite
- **Software Testing (Chapter 8)**: This chapter gives a briefing about testing for various modules
- **Result (Chapter 9)**: This chapter discusses about the output and snapshots of the project.
- **Deployment and Maintenance (Chapter 10)**: This chapter summarizes the deployment and maintenance required for the project.
- **Conclusion and Future Scope (Chapter 11)**: This chapter summarizes and concludes the project report and give the future scope.
- **Plagiarism Report(Chapter 12)**: This chapter shows the plagiarism report.

Chapter 2

Literature Survey

The growing adoption of gesture recognition and voice technology in digital presentations is evident in the literature. These technologies aim to enhance user engagement, accessibility, and interactivity, particularly in educational and professional settings. While machine learning techniques have improved the precision and adaptability of these systems, challenges remain. Issues such as misclassification of gestures, environmental sensitivity, and dependence on specific hardware continue to impact their effectiveness. Despite these challenges, the research indicates that there is significant potential for hands-free, intelligent presentation tools, paving the way for more immersive and accessible experiences in the future.

R. M. Shakya, S. P. Sharma, and N. Shrestha, "The Use of Hand Gestures as a Tool for Presentation," International Journal of Advanced Computer Science and Applications, vol. 14, no. 11, 2023:

This study investigates the use of hand gestures as an alternative to traditional presentation control methods. It employs machine learning models to perform real-time gesture recognition and has been tested in environments like classrooms and corporate settings. The results suggest that the system enhances user engagement and accessibility, particularly benefiting individuals with disabilities by providing a hands-free control option. However, the system faces challenges, such as misinterpreting complex gestures and difficulties in low-light conditions, indicating that refining the recognition algorithms could improve accuracy in varied environments.[1].

S. R. Patel, "Gesture Recognition: Applications in Education and Presentations," IEEE Transactions on Human-Machine Sys- tems, vol. 53, no. 3, pp. 250-260, 2023:

This paper explores the use of gesture recognition technology in educational and

presentation settings, aiming to create hands-free, interactive environments. The system facilitates presenters in controlling their content through natural gestures, thereby enhancing user engagement and creating a more immersive experience. Nonetheless, the reliance on consistent environmental conditions for accurate gesture recognition may limit the system's effectiveness in diverse settings.[2]

M. K. Jha and A. D. Gupta, "A Study on Gesture Recognition in Interactive Presentations," ITM Conference Proceedings, ICACC 2022:

This study examines gesture recognition technologies, specifically combining convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to improve real-time gesture detection in presentations. The hybrid model demonstrated better accuracy in detecting gestures across varying conditions, making it more adaptable to different environments. However, the system struggled with very fast or subtle gestures, occasionally misclassifying them.[3].

A. Lee and B. Chan, "Enhancing Presentations with Gesture Recognition Technologies," IEEE, vol. 10, pp. 500-510, 2022:

This research integrates gesture recognition technologies into presentation systems, evaluating their effectiveness in professional and educational contexts. The system, which also supports voice commands and multimedia interaction, was shown to improve user engagement by allowing seamless, hands-free control of presentations. However, limitations include occasional difficulty with complex gestures and environmental factors[4].

J. P. Brown, "The Future of Interactive Presentations: Gesture and Touch," IEEE Transactions on Education, vol. 65, no. 2, pp. 120-130, 2022:

This research investigates the combination of gesture and touch technologies for interactive presentations. By blending these two input modalities, the system enhances user interaction and enables multi-user collaboration. While the findings indicate that this combination makes presentations more intuitive, the challenge of differentiating between multiple users' gestures remains[5]

L. Zhang, "Utilizing Kinect for Enhanced Presentation Control," IEEE Transactions on Multimedia, vol. 24, pp. 1-10, 2022:

This study examines the use of Kinect for gesture-based presentation control, leveraging its depth-sensing technology for more accurate gesture detection. The findings demonstrate Kinect's effectiveness in low-light environments, though the system's high dependency on specific hardware and lack of portability are limitations[6]

S. Powar, S. Kadam, S. Malage, and P. Shingane, "Automated Digital Presentation Control using Hand Gesture Technique," 2022:

This research focuses on automating digital presentation control through hand gestures, eliminating the need for physical input devices. The developed system uses computer vision algorithms to detect hand gestures, significantly improving user convenience. However, accuracy is limited by environmental factors, suggesting the need for optimized hardware and refined recognition algorithms[7]

O. Mubin, J. Henderson, and C. Bartneck, "A Comparative Analysis of Real-Time Open-Source Speech Recognition Tools for Social Robots," 2020:

This research evaluates various open-source speech recognition tools for social robots, analyzing real-time performance, accuracy, and adaptability. Many tools perform well in isolated speech commands but struggle with noisy environments or multiple speakers, indicating the need for robust systems in interactive settings[8]

T. Kirishima, K. Sato, and K. Chihara, "Real-Time Gesture Recognition by Learning and Selective Control of Visual Interest Points," March 2005:

This research presents a method for real-time gesture recognition by focusing on selective visual interest points in video sequences. The system uses machine learning techniques to detect and track these points, enabling accurate recognition of dynamic hand gestures[9]

College of Computer and Cyber Security, Communication Uni- versity of China, "Video Subtitle Location and Recognition Based on Edge Features," 2022:

This study introduces a method for locating and recognizing subtitles in video frames using edge detection techniques. The findings show that edge feature analysis allows for efficient subtitle extraction even in videos with complex backgrounds. However, limitations were noted when dealing with fast-moving text or videos with low contrast between the text and background. The study suggests that integrating additional image processing techniques could improve the system's robustness in challenging conditions[10]

2.1 Conclusion From Literature Survey

The literature survey highlights the advancements in gesture recognition technology for enhancing user engagement in presentations. While gesture-based control provides a hands-free experience, challenges such as environmental dependency, complexity in gesture recognition, and hardware limitations persist. Integrating machine learning tech-

niques shows promise for improving accuracy and adaptability, but further refinement is needed to enhance reliability across diverse conditions. Future research should aim to develop more flexible and robust systems for broader device compatibility, ultimately transforming presentation delivery through improved user interaction.

Chapter 3

Software Requirement Specification

3.1 Introduction

3.1.1 Purpose

The purpose of this document is to define the objectives and scope of the AI-based Presentation Tool that uses gesture recognition and voice commands to enable hands-free control of presentations. It outlines the system's key features, including real-time gesture detection, speech integration, customizable slide control, and offline functionality. The tool is designed to enhance user experience by making presentations more interactive, accessible, and intuitive, especially for educators, professionals, and speakers. It also emphasizes user privacy, ease of use, and the integration of AI-powered features like speech-to-text and real-time translation to ensure inclusivity and engagement across diverse audiences.

Why the project is Chosen?

This project was chosen to solve the common limitations faced during presentations, such as the need to manually operate devices or stay near a computer to change slides. By using gesture recognition and voice commands, the tool allows presenters to move freely and interact naturally with their content. It aims to make presentations more engaging, especially in classrooms, conferences, and professional settings. The idea also aligns with current trends in AI and human-computer interaction, making it both innovative and practically useful.

3.1.2 Intended audience and reading suggestion

- **Educators and Teachers:** Can use the tool in classrooms and seminars to deliver more dynamic and interactive presentations without needing to stay near a computer or use a remote.

- **Corporate Professionals:** Useful during meetings, pitches, and conferences where smooth, hands-free slide control can enhance the flow and professionalism of the presentation.
- **Public Speakers and Trainers:** Enables freedom of movement and natural interaction with the audience, improving engagement and communication.

3.1.3 Project Scope

The scope of this project includes the design and development of an AI-based presentation tool that enables hands-free slide control using gesture recognition and voice commands. The system will be capable of detecting predefined gestures in real-time through a webcam and executing corresponding actions such as next slide, previous slide, pause, or interact. It will also support voice commands for additional flexibility. The tool is intended to work offline and be compatible with commonly used presentation software. Key features include customizable gestures, speech-to-text subtitles, real-time translation, and an intuitive user interface. The project focuses on improving accessibility, user engagement, and interaction during presentations for a wide range of users including educators, professionals, and students.

3.1.4 Design and Implementation Constrain

1. The gesture recognition system will rely on camera input processed using the MediaPipe framework.
2. The desktop application will be developed using Python with GUI frameworks like Tkinter or CustomTkinter.
3. The system must be compatible with commonly used presentation software such as Microsoft PowerPoint.
4. Voice command processing will be limited to predefined actions to ensure accuracy and speed.
5. Developers and administrators must have access to complete technical documentation for maintenance and future upgrades.

3.1.5 Assumption and Dependencies

1. The user's device must have a functioning webcam and microphone for gesture and voice recognition.

2. The system assumes a stable lighting environment to accurately detect hand gestures through the camera.
3. Users will perform predefined gestures correctly within the frame for the system to respond accurately.
4. Speech recognition features assume clear audio input without excessive background noise.
5. The tool assumes that the host system has the necessary Python environment and dependencies installed.

3.2 System Features

3.2.1 Gesture Control

The system allows users to navigate through slides and control various presentation elements using simple hand gestures. This hands-free approach enhances the ease of use and makes presentations more fluid and professional.

3.2.2 Voice Commands

With voice commands, presenters can transition between slides and add annotations effortlessly. This feature offers a seamless way to interact with the presentation without interrupting the flow.

3.2.3 Customization Options

Users can tailor the tool to their unique preferences by personalizing gestures. This flexibility ensures that the system adapts to individual styles and enhances the user experience.

3.2.4 Speech-to-Text

Real-time on-screen subtitles are generated as the presenter speaks. This feature not only enhances accessibility but also ensures that the content is clear and easy to follow for all audience members.

3.2.5 Real-Time Translation

To support a multilingual audience, the tool provides real-time translation of the presenter's speech into multiple languages. This feature promotes inclusivity and makes the presentation accessible to a global audience.

3.3 External Interface Requirement

3.3.1 User Interface

The user interface will be designed to be simple, responsive, and user-friendly, allowing presenters to easily configure gestures and voice commands before starting their presentation. The main screen will include controls to launch presentation mode, set preferences, and customize gestures. Visual feedback will be provided to confirm recognized gestures and voice inputs during live presentations.

3.3.2 Software Interface

The software interfaces of the AI Presentation Tool involve both frontend and backend components, integrated seamlessly using a variety of libraries and APIs. The frontend is built using Python's Tkinter library, which provides an intuitive GUI where users can enable or disable features like gesture control, voice commands, subtitles, and language preferences. The backend logic is powered by Python 3.x, handling data flow and feature execution. The tool leverages OpenCV to capture and process webcam video streams, while MediaPipe is used to detect hand gestures by identifying 21 landmark points on each hand. PyAutoGUI translates these gestures into keyboard shortcuts for controlling slide navigation. For voice functionalities, Vosk is employed to convert speech into real-time subtitles, and Google Text-to-Speech (gTTS) is used to interpret and process voice commands. Additionally, the Deep Translate library supports real-time translation of subtitles into multiple languages, such as Hindi and English. These software interfaces collectively ensure that the tool operates smoothly, accurately, and in real time.

3.3.3 Communication Interface

The communication interfaces in the AI Presentation Tool are primarily internal, as the system is designed to function offline for maximum reliability during live presentations. The interaction between different modules—such as gesture recognition, voice processing, and GUI controls—is handled locally within the Python environment. Libraries like MediaPipe and OpenCV process real-time video feed from the webcam, while Vosk handles voice input without the need for internet connectivity. If optional features like real-time subtitle translation are enabled, the Deep Translate library can operate using offline models or minimal internet access, depending on configuration. All components are integrated securely and efficiently to ensure smooth data flow and real-time responsiveness without relying on cloud-based services or external servers.

3.4 Non Functional Requirements

1. **Performance:** Ensure real-time gesture and voice command recognition with minimal latency during presentations.
2. **Usability:** Provide an intuitive and user-friendly GUI using Tkinter for easy interaction, even by non-technical users.
3. **Reliability:** Operate smoothly in offline mode without internet dependency, ensuring uninterrupted performance.
4. **Compatibility:** Support standard hardware such as webcams and microphones across various screen resolutions.
5. **Portability:** Capable of running on different operating systems that support Python and its dependencies.

3.5 Other Requirement

3.5.1 Database Requirements:

The system will use MySQL as the backend database to store and manage user-related information and custom gesture mappings. The database will include tables for user profiles, custom gesture definitions, user preferences, and usage history. Each user will have a unique ID to associate their personalized settings and gestures. The gesture table will store predefined gestures along with corresponding keyboard actions or commands. The database must support fast read/write operations to ensure real-time responsiveness when fetching gesture configurations during presentations. Proper indexing and relational integrity will be maintained to ensure data consistency and scalability. Additionally, basic encryption or access controls can be applied to protect sensitive data such as user profiles and personalized settings.

3.6 Analysis Model

3.6.1 Data Flow Diagram

DFD Level 0

The Level 0 Data Flow Diagram (DFD) of the AI Presentation Tool outlines the core system and its interactions with external components. The central process, "AI Presentation Tool," receives input from the user through gestures and voice, which are processed via the Gesture Detection Module and Voice Command Module. Based on these inputs, the system controls the flow of presentation slides and communicates feedback to the user. It also interacts with a MySQL Database to store and retrieve user preferences and custom gestures. Additionally, the system sends spoken input to the Subtitle Translation Module to generate real-time translated subtitles, ensuring accessibility and enhancing the presentation experience.

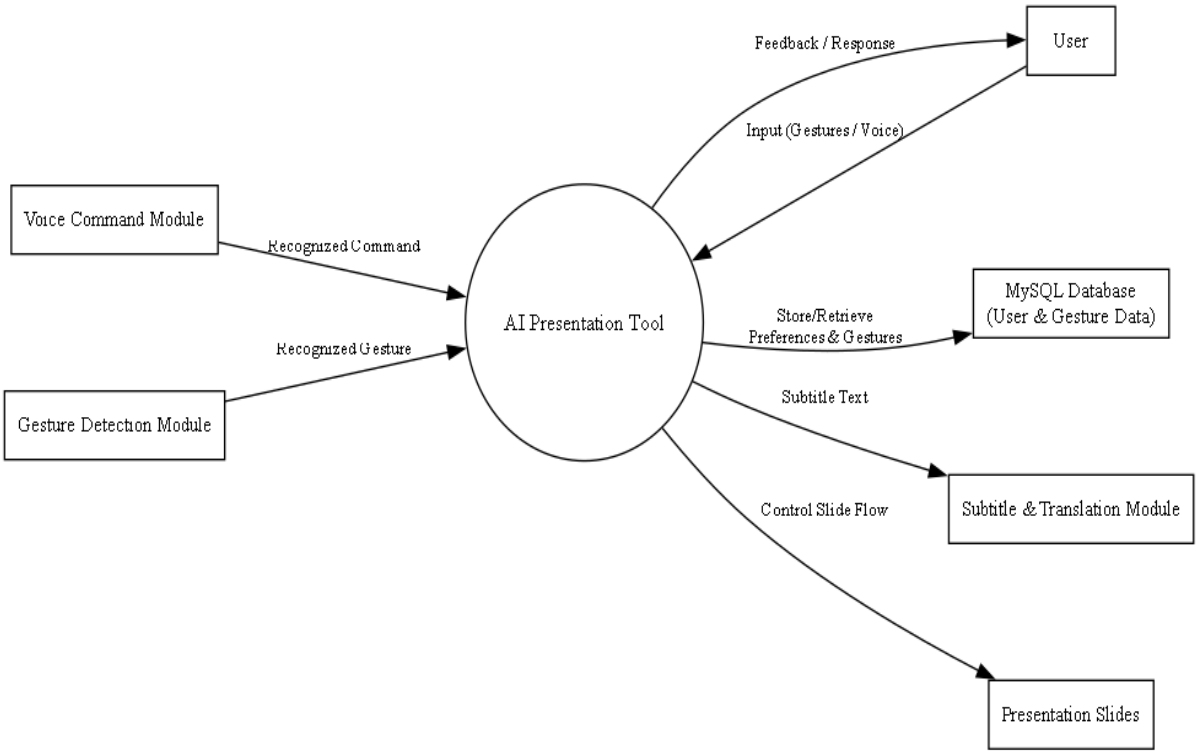


Figure 3.1: Dataflow Diagram

DFD Level 1

The Level 1 Data Flow Diagram (DFD) for the AI Presentation Tool illustrates the interaction between the User, core system processes, and the Data Store. The user provides input via Voice Commands and Gestures, which are captured and processed by the system. The Voice Command Process handles spoken commands, converting them

into executable actions, while the Gesture Recognition Process identifies and analyzes user gestures for slide navigation or interactive control.

To enhance accessibility, the Real-Time Translation Process translates spoken or written content into different languages, ensuring global usability. The Speech to Text Process transcribes speech into subtitles, improving comprehension and inclusivity during presentations. Additionally, the Gesture Customization Process allows users to define personalized gestures, which are saved in the Data Store for future use.

The central System integrates these processes, executing commands and providing feedback to the user. All relevant data, such as subtitles, translations, and custom gestures, is stored and retrieved from the Data Store, ensuring seamless interaction and personalization throughout the presentation. This DFD emphasizes the system's modular and user-centric design, which enhances engagement and accessibility.

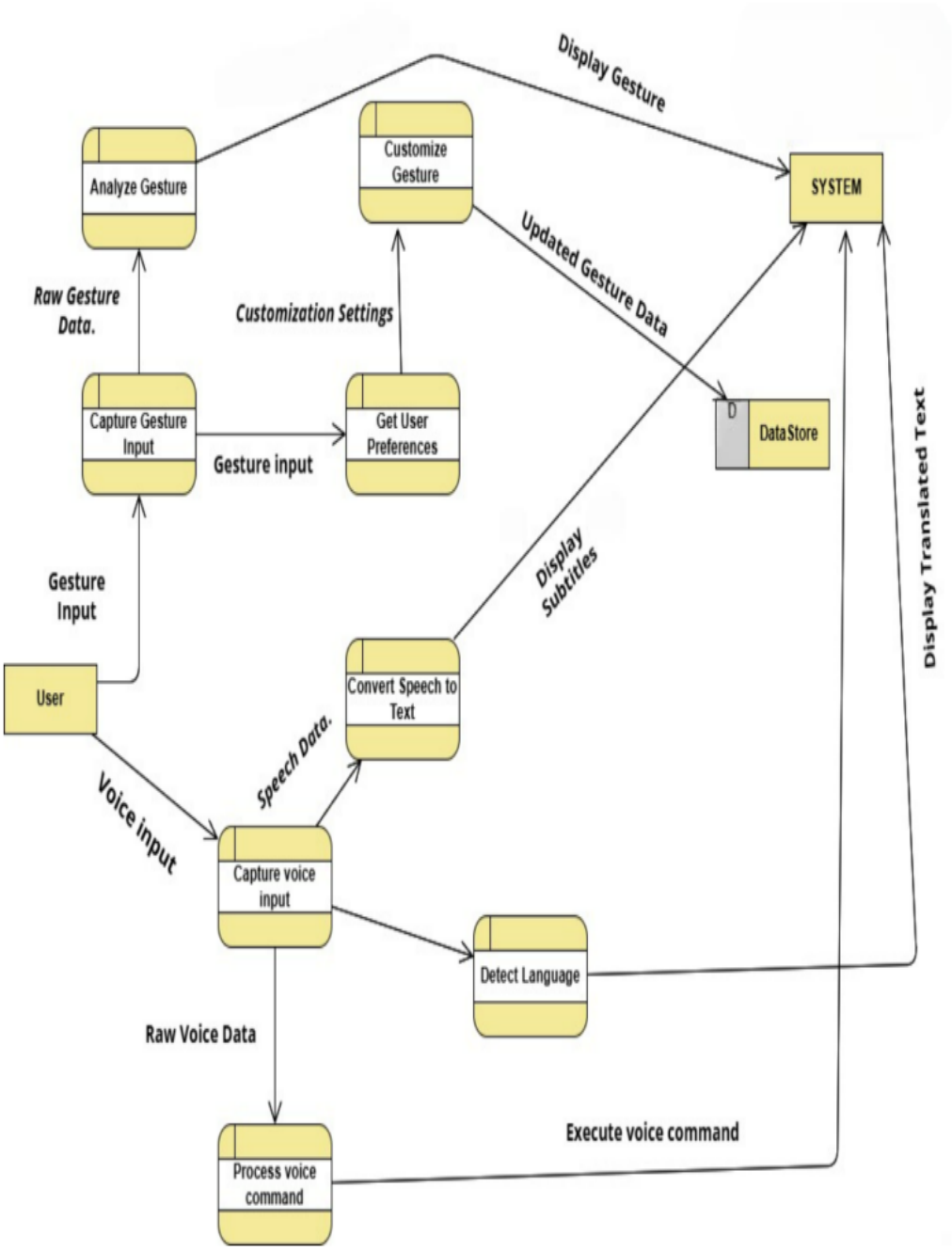


Figure 3.2: Dataflow Diagram

3.6.2 State Machine Diagram

The State Diagram for the AI Presentation Tool provides an overview of how the system transitions between different states to deliver a seamless and interactive user experience. The process begins in the Start state, where the system initializes all necessary components. Once ready, it moves to the Idle state, waiting for user input to proceed further.

When a user selects a presentation, the system enters the In Preparation state, loading slides and necessary configurations. Once everything is set up, the tool transitions to the Ready to Present state, signaling that it is prepared to start the presentation. Upon user confirmation, the system shifts to the Presenting state, where the core functionalities like navigating slides and responding to voice commands or gestures are actively monitored.

During the presentation, users can pause the session, placing the system in the Paused state. If real-time subtitles are enabled, the system enters the Subtitles state, providing on-screen captions to enhance accessibility. Additionally, the system can transition to the Interacting state, allowing users to interact with multimedia elements like videos, images, and charts. Finally, once the presentation concludes, the system moves to the End state, saving any updates and returning to Idle, ready for the next session. These transitions ensure a smooth flow of control, offering a hands-free and interactive experience tailored to various user needs.

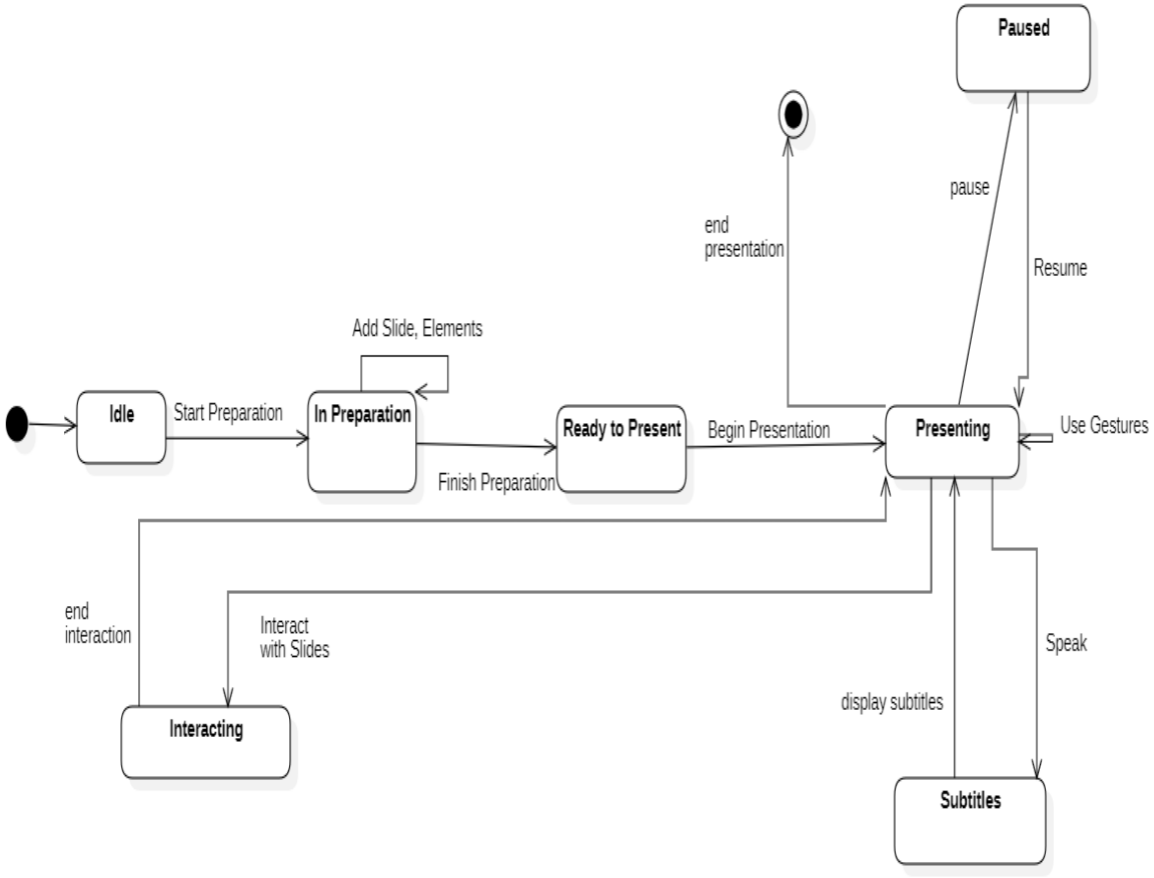


Figure 3.3: State Machine Diagram

3.7 System Implementation Plan

3.7.1 Implementation Plan

Task No.	Task to be Accomplished
T1	Topic Finalization
T2	Requirement specification
T3	Technology Familiarization
T4	System Set up
T5	Concept Review Study
T6	Study of technologies used in the project
T7	Design of user interface
T8	Designing of Constraints Rules
T9	Creation of Connectivity for Services
T10	Creation of database files and rules
T11	Designing the Architectural layout
T12	Creating module using database and rules
T13	Integration of T11 to T7 Tasks
T14	Testing
T15	Documentation Preparation
T16	Maintenance

Table 3.1: Implementation Plan

Chapter 4

System Design

4.1 System Architecture

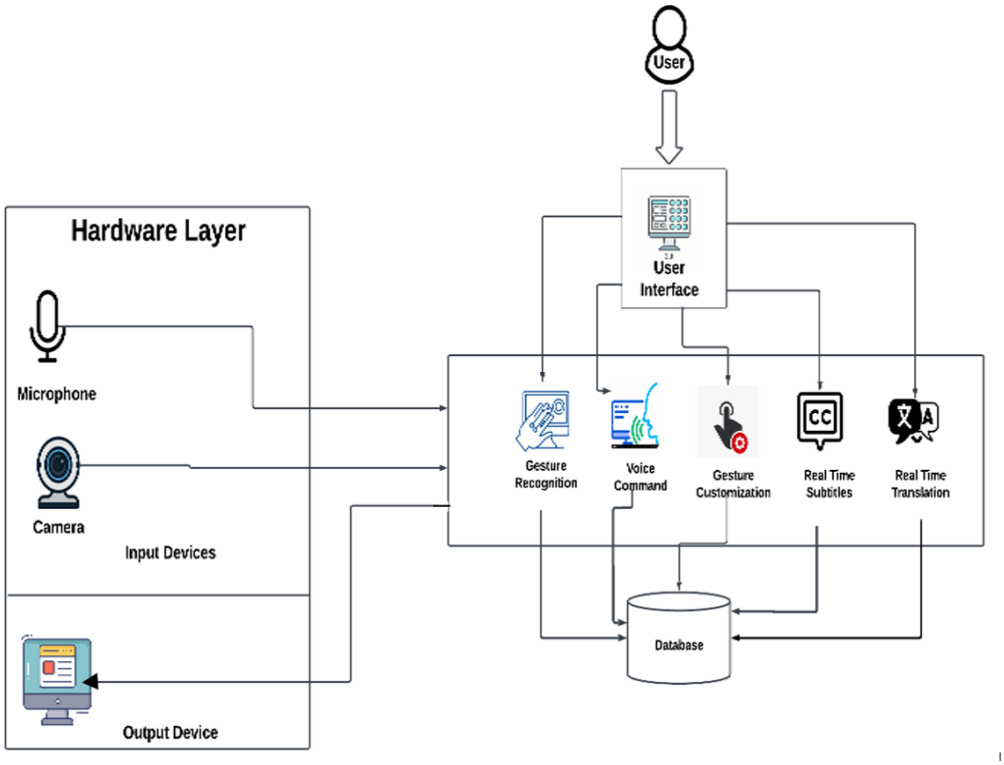


Figure 4.1: System Architecture

The architecture of the AI Presentation Tool is structured into four key layers: User Interface Layer, Application Logic Layer, Data Management Layer, and Hardware Layer. Together, these layers enable real-time, hands-free control of presentations through gestures and voice commands.

1. User Interface Layer: Built with Tkinter, this layer offers a user-friendly interface that allows users to configure gesture, voice, and subtitle preferences, control slide

navigation, and customize gestures and voice commands for specific presentation actions.

2. Application Logic Layer: This core layer integrates key modules such as Gesture Recognition (MediaPipe) for mapping hand gestures to actions, Voice Commands (SpeechRecognition) for executing spoken instructions, Gesture Customization for personalized controls, Real-Time Subtitles for transcribing speech, and a Translation Module for converting speech into multiple languages.

3. Data Management Layer: This layer manages the storage of user profiles, gesture definitions, and voice command mappings, ensuring a personalized experience and supporting future improvements through data analytics.

4. Hardware Layer: This layer includes input devices like a camera for gestures and a microphone for voice, along with output devices such as a display for slides and subtitles, and a computing system to process all actions in real time.

4.1.1 Working :

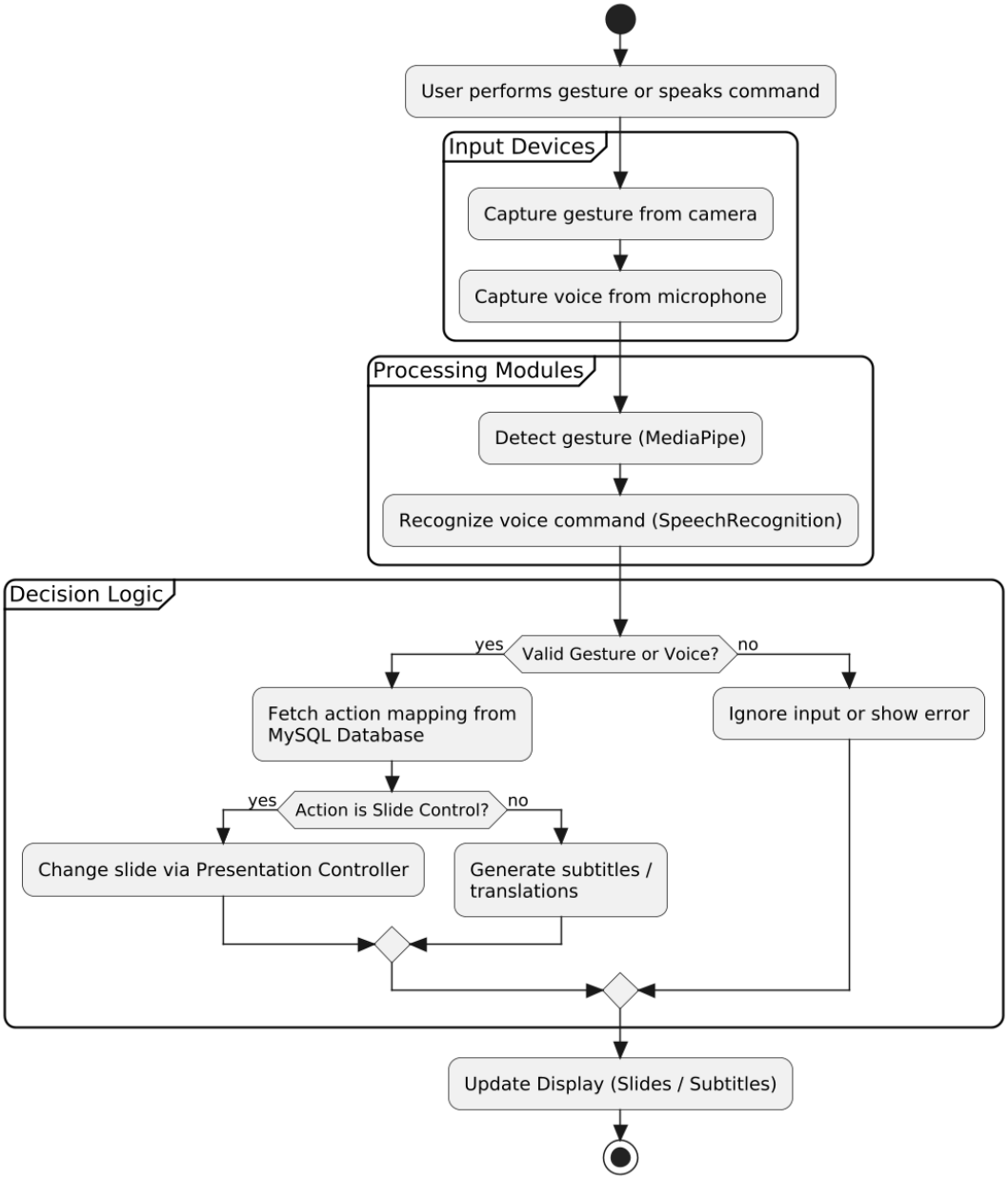


Figure 4.2: Working

The flowchart outlines the working mechanism of the AI Presentation Tool, starting with user interaction. The user either performs a gesture or speaks a command, which is captured by input devices—a camera for gestures and a microphone for voice input. These inputs are then processed by specialized modules: the Gesture Detection Module (using MediaPipe) interprets hand movements, while the Voice Command Module (using SpeechRecognition) converts spoken words into actionable commands.

Once the inputs are recognized, the system checks if they are valid and mapped to any predefined action. If valid, the system queries the MySQL database to retrieve

the specific mapping for that input. Depending on the type of action, the tool either controls the slide presentation (e.g., next, previous, pause) or passes the voice data to the Subtitle Translation Module to generate real-time subtitles or translations.

Finally, the output—either an updated slide or subtitles—is displayed to the user through the output screen, completing the real-time, interactive loop of the AI-based presentation control system.

4.2 UML Diagrams

4.2.1 Entity Relationship Diagram

The ER diagram models the data structure for your AI Presentation Tool that uses gesture recognition. It consists of two primary entities: Users and Gestures. Each user has a unique id, a username, and a password, which helps in authentication and user-specific customization. The Gestures entity captures information about gestures defined by the users. Each gesture has its own id, the action it performs (such as next slide or pause), the gesturedata (which could be encoded movement data or labels), and an optional imagepath representing a visual reference for the gesture. There is a one-to-many relationship between Users and Gestures — meaning a single user can define multiple gestures, but each gesture is linked to one specific user. This design allows the system to store personalized gesture-action mappings per user, enabling tailored control during presentations.

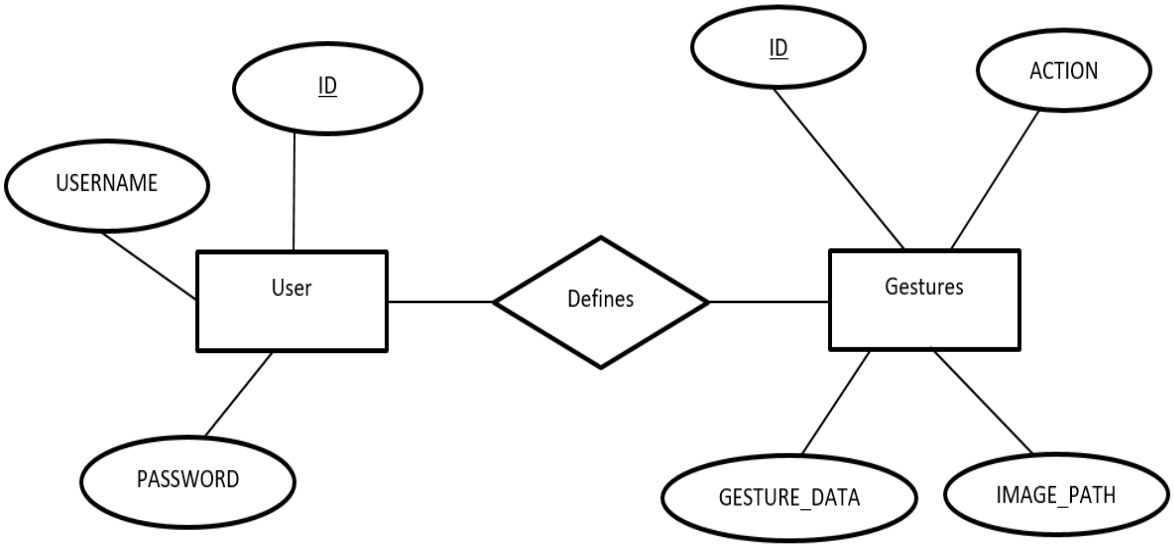


Figure 4.3: ER Diagram

4.2.2 Activity Diagram

The process begins with the user starting the application. Upon launch, the system performs essential initialization steps, including activating hardware components like the webcam and microphone, and loading necessary libraries for gesture recognition and voice processing. This ensures that the system is ready to receive real-time input from the user.

Once initialization is complete, the system enters an idle state where it constantly listens for inputs. The user can either perform a gesture or speak a command. At this point, the system determines the type of input: if a gesture is detected, the system captures video frames using the webcam, processes them through a gesture recognition module (such as MediaPipe), and classifies the gesture using pre-trained models. If a voice command is given instead, the system records the audio input, converts it to text using speech-to-text technology, and analyzes the command using natural language processing or keyword detection.

Based on the interpreted input—gesture or voice—the system proceeds to an action lookup phase. Here, it maps the interpreted input to a predefined action like "Next Slide", "Previous Slide", "Start Presentation", or "End Presentation". If a valid action is identified, the system executes the command, updating the presentation state and optionally displaying real-time subtitles or translations as feedback. In case the system cannot match the input to a known command, it informs the user by displaying an error message or requesting a repeat input. The diagram concludes with the system either successfully completing the action or providing appropriate feedback, returning to the listening state for continuous interaction

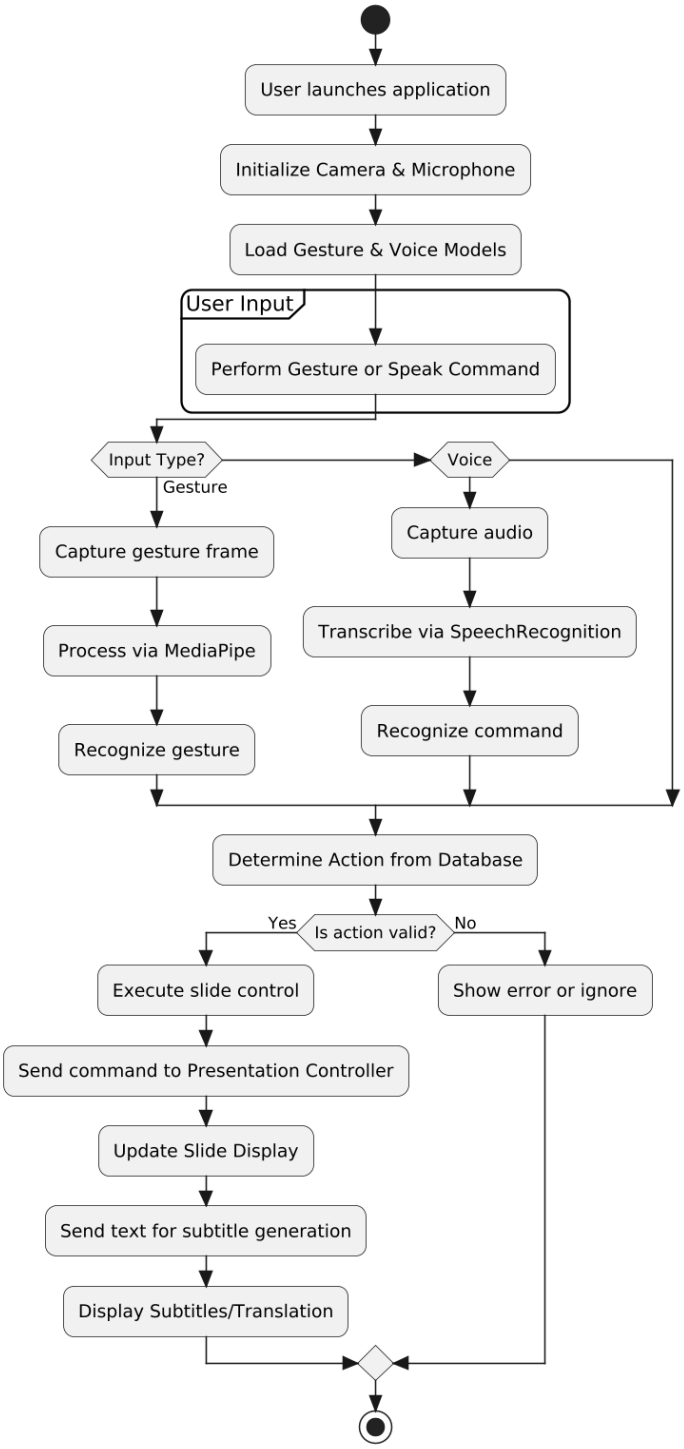


Figure 4.4: Activity Diagram

4.2.3 Use Case Diagram

This use case diagram represents an AI-Based Presentation System Using Gesture Recognition, where two primary actors interact with the system: the Presentator and the System. The Presentator can initiate the presentation through the "Start Presentation" use case, which includes actions like "Navigate Slides" and "Control Interactive Elements" to manage the flow and interaction of the presentation. Users can also "Customize Gestures" to tailor the system to their preferred motions. The system supports "Give Voice Commands," enabling control through spoken input. Additional features include the ability to "Display Subtitles" and "Provide Real-Time Translation," both of which are extended use cases triggered by the system to enhance accessibility and audience understanding. Overall, the diagram illustrates a smart, interactive presentation system focused on intuitive control through gestures and voice, supported by real-time AI-powered feedback and enhancements.

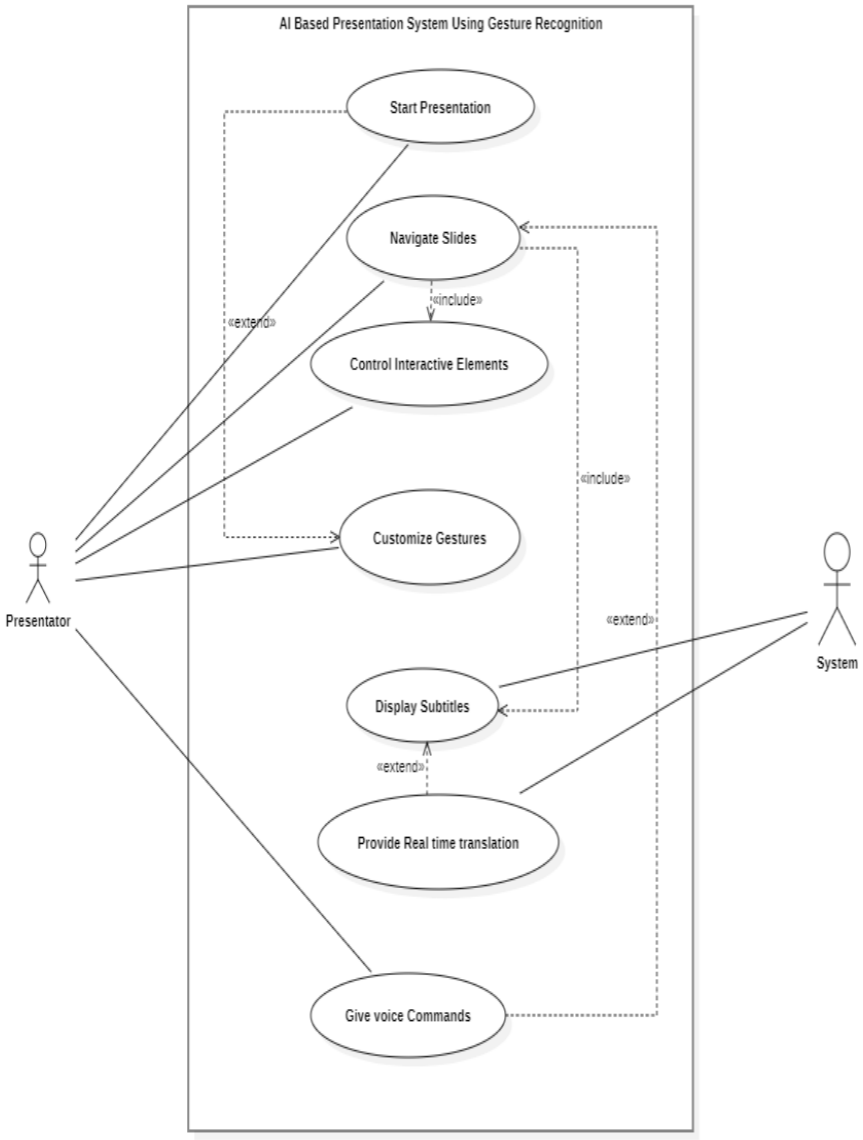


Figure 4.5: Use Case Diagram

4.2.4 Sequence Diagram

The Sequence Diagram represents how key components of the AI Presentation Tool interact during a typical presentation session. It begins with the Presenter initiating the session by sending a start request to the PresentationManager, which then loads the first slide. Depending on the input method used, either the Voice Command or Gesture Recognition module is activated. When the Presenter gives a voice command like "Next Slide," the Voice Command module processes it and instructs the PresentationManager to update the slide. The PresentationManager then sends the new content to the UI for display.

Alternatively, if the Presenter uses gestures, the Gesture Recognition module detects the motion and sends a corresponding instruction to the PresentationManager. The manager again updates the slide and forwards the content to the UI. Throughout the session, the UI handles rendering slides and interactive elements, ensuring smooth transitions and display. In essence, the Presenter interacts using voice or gestures, the system processes inputs via specialized modules, and the PresentationManager coordinates everything while the UI delivers a seamless visual experience.

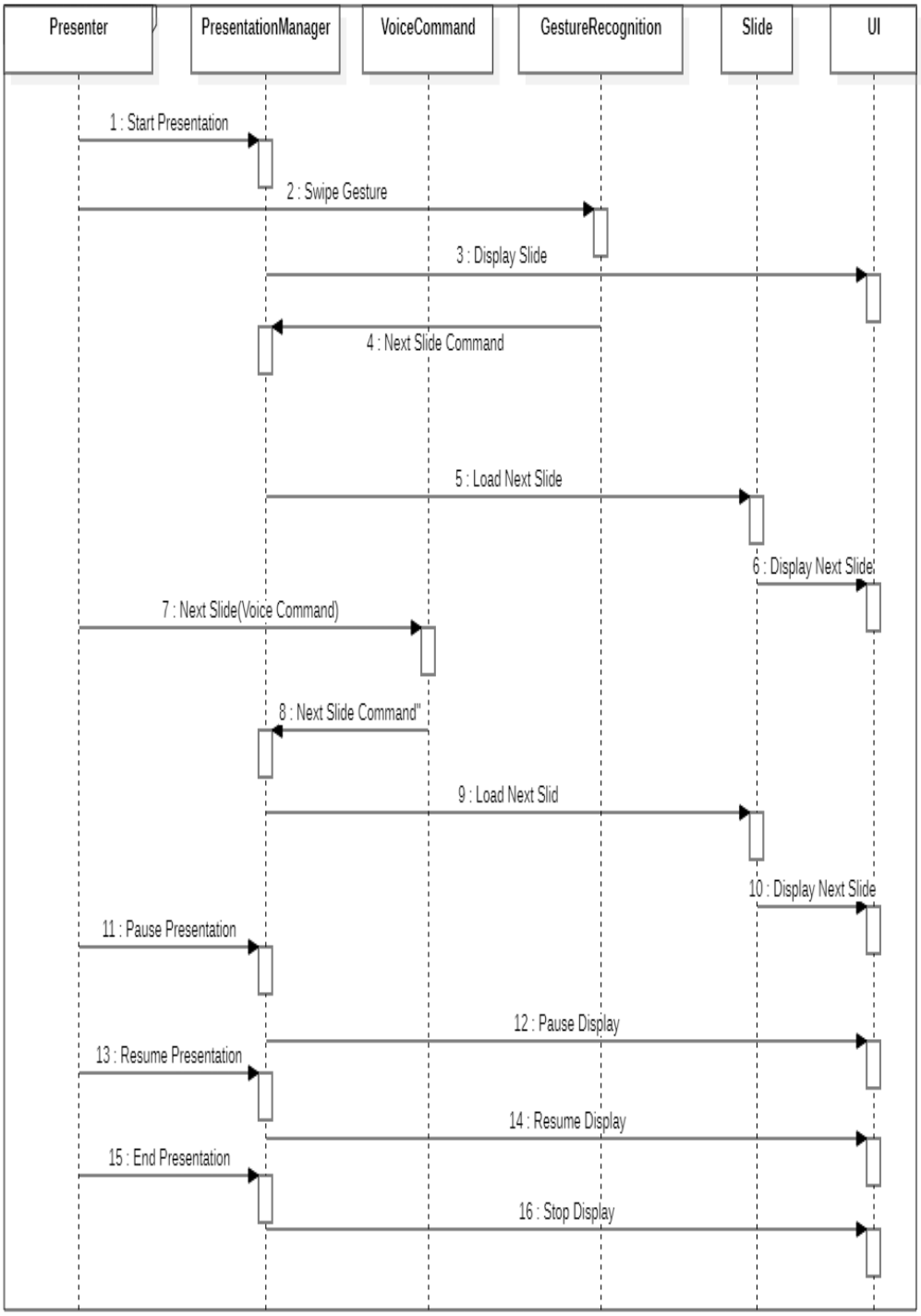


Figure 4.6: Sequence Diagram

4.2.5 Component Diagram

The component diagram illustrates the architecture of the AI Presentation Tool system. The core "AI Presentation Tool" component manages the overall presentation logic and interacts with the "User Interface" for displaying content and receiving direct user input. User interaction is enhanced through dedicated "Gesture Recognition" and "Voice Command Processing" components, which interpret physical movements and spoken instructions, respectively, and feed these inputs back to control the main presentation tool. The "Voice Command Processing" component further interacts with a "Real-Time subtitles" module to generate captions from the detected speech. This subtitles module can then pass the text to a "Real-Time Translation" component for providing multilingual support. A central "Database" serves as a repository, used by Gesture Recognition, Real-Time subtitles, Real-Time Translation, and the User Interface, likely for storing gesture data. The diagram also indicates an interaction between Gesture Recognition and Voice Command Processing for synchronizing or combining multimodal inputs.

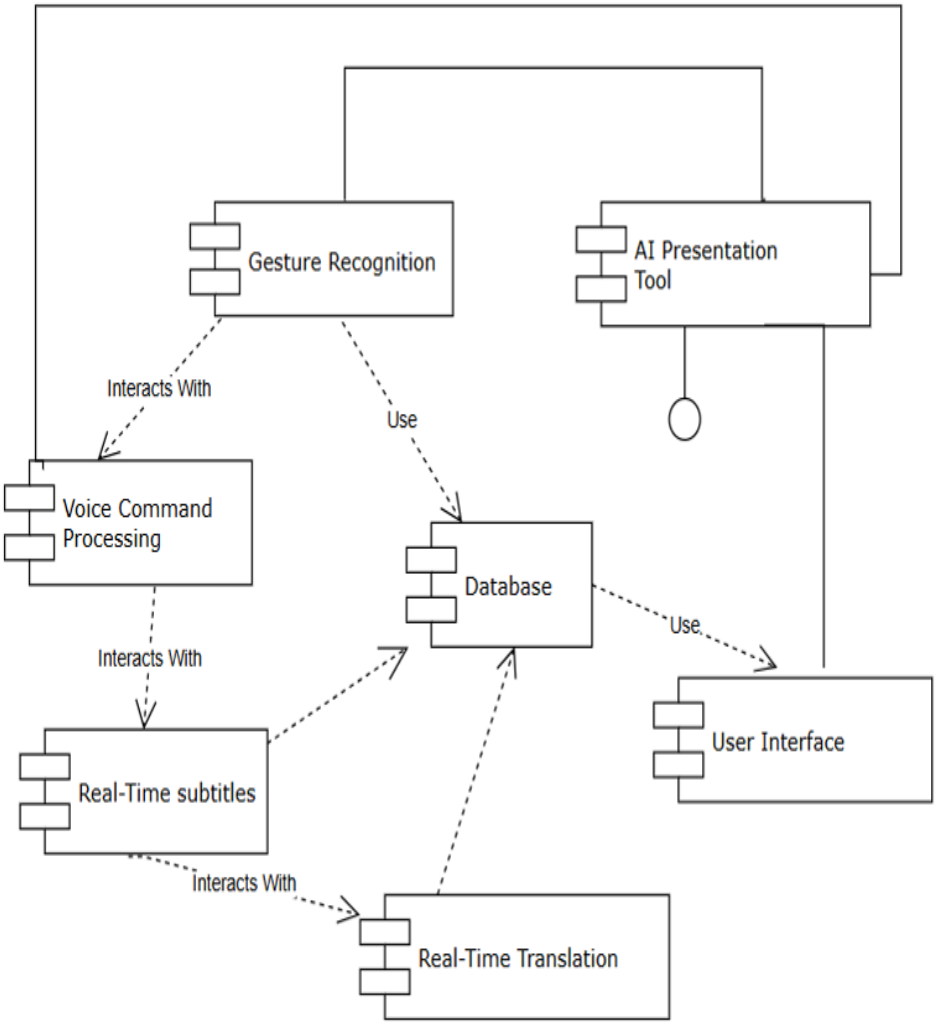


Figure 4.7: Component Diagram

4.2.6 Deployment Diagram

The deployment diagram details the runtime components of the AI Presentation Tool. User interaction occurs via gestures and voice commands, which are captured and processed by dedicated "Gesture Recognition" (using MediaPipe/OpenCV) and "Voice Recognition" (using Speech APIs like VOSK/Google) components, respectively. These input components feed into the core "Application Logic," implemented in Python, which serves as the central processing unit of the system. This logic component utilizes a "Database" (employing SQL) for data storage and retrieval. For user interaction and control, the "Application Logic" drives a "Graphical User Interface" built with Tkinter. The final presentation output, including slides and multimedia control, is managed by the "Output Interface" component, also directed by the "Application Logic."

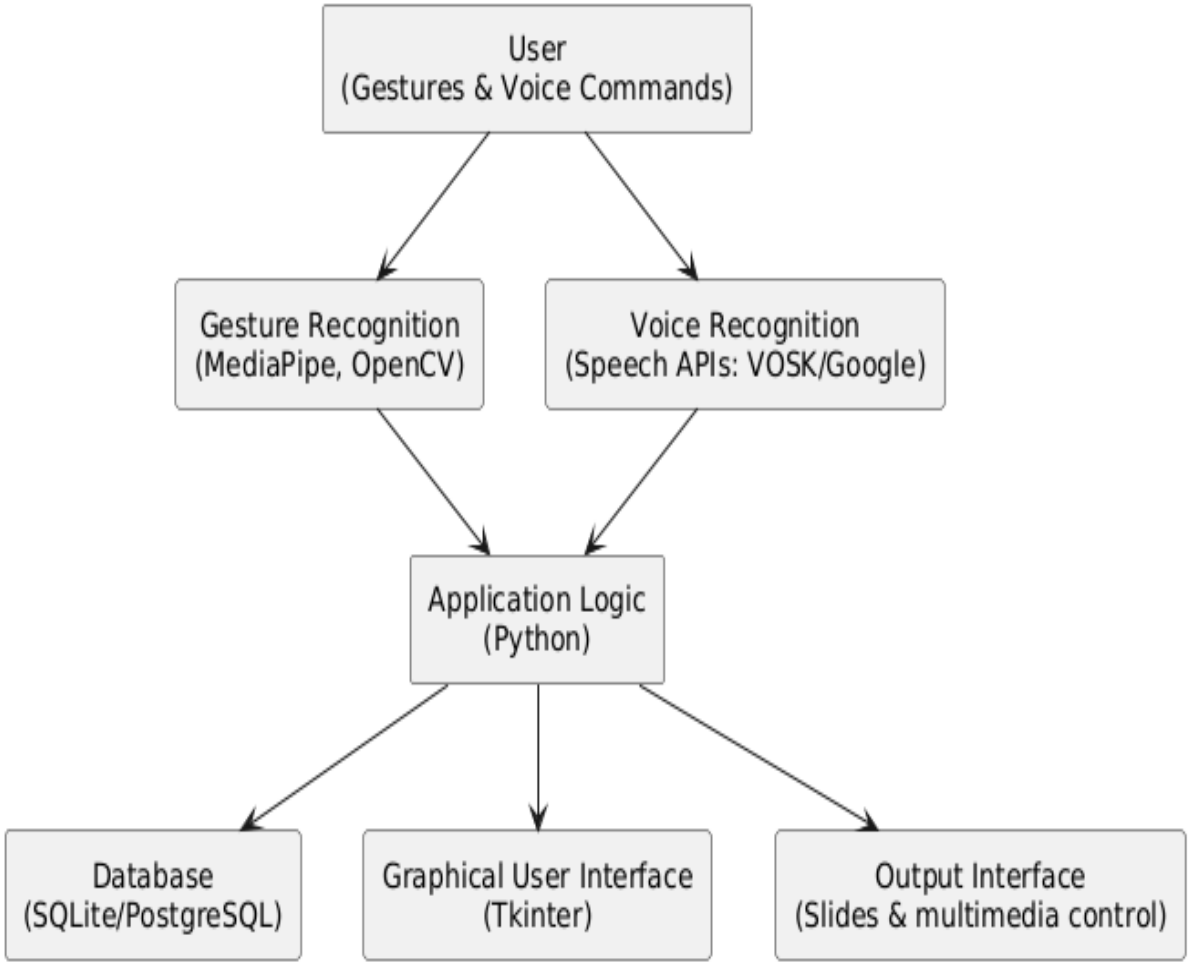


Figure 4.8: Deployment Diagram

Chapter 5

Technical Specifications

5.1 Technology details used in the project

For the development of our AI-based Presentation Tool using Gesture Recognition, we have utilized a combination of powerful and efficient technologies to ensure a seamless, offline-capable, and real-time interactive experience. The core technologies include Python for backend processing, OpenCV and MediaPipe for gesture detection, Tkinter/CustomTkinter for the desktop GUI, and Text-to-Speech (TTS) libraries for auditory feedback. Each component plays a critical role in delivering a smart, hands-free presentation system. Below is an overview of the technologies used :

1. **Python:** Python serves as the main programming language for this project due to its readability, extensive libraries, and strong support for machine learning and image processing tasks. Python's modularity and wide community support make it ideal for implementing AI-powered features such as gesture recognition, slide control, and voice feedback.

2. **MediaPipe:** Developed by Google, MediaPipe is a cross-platform framework for building multimodal machine learning pipelines. In our project, MediaPipe is primarily used for real-time hand gesture detection. It provides highly accurate landmark tracking, allowing us to interpret hand movements as slide navigation commands like "Next," "Previous," or "Pause."

3. **OpenCV:** OpenCV (Open Source Computer Vision Library) complements MediaPipe by handling image acquisition, pre-processing, and rendering the webcam feed. It enables real-time visual input and supports the overlay of detected gesture data for user feedback and debugging.

4. **Tkinter / CustomTkinter:** For the desktop user interface, we used Tkinter and its enhanced version, CustomTkinter, to design a simple yet effective GUI. This includes controls for starting/stopping the presentation, toggling between input

modes (voice or gesture), and displaying current slide status, making the tool user-friendly and intuitive.

5. **SpeechRecognition:** Captures the presenter's voice and converts speech into text. Used for real-time subtitle generation during the presentation

6. **DeepTranslator:** Translates the live transcribed speech into different languages.Supports real-time translation for multilingual accessibility.

By combining these technologies, our system supports an interactive, hands-free presentation experience tailored for professionals, educators, and speakers. MediaPipe and OpenCV together ensure accurate gesture control, while Python's ecosystem handles real-time logic and system integration. The use of Tkinter and TTS modules enhances usability and interactivity.

The modular design also ensures future scalability—features like voice commands, speech-to-text subtitles, and real-time translation can be integrated easily thanks to Python's vast support for AI and NLP libraries. This flexible and efficient tech stack enables us to build a reliable, responsive, and innovative AI-powered presentation system.

Chapter 6

Project Estimation Schedule and Team Structure

6.1 Project Estimate

COCOMO Model The Constructive Cost Model (COCOMO) is a well-known method for estimating software project costs. Developed by Barry Boehm, it offers a structured approach to estimate the work, time, and cost required to develop a software project. COCOMO includes three versions: Basic, Intermediate, and Advanced, each building on the previous version with more detailed factors considered for project estimation.

1. **COCOMO Basic:** In the Basic model, the software development effort is estimated based on the project's size, measured in lines of code (LOC). The formula for estimating effort (in person-months) is:

$$Effort = a * (KLOC)^b$$

   where,

   - The total development effort is measured in person-months as effort.
   - KLOC stands for thousand lines of code, which is a measure of the software's projected size.
   - The constants a and b depend on the experience of the development team and the type of project (such as organic, semi-detached, or embedded).

2. **COCOMO Intermediate:** The Intermediate model builds on the basic model, incorporating additional project characteristics such as product quality, hardware limitations, human resource capabilities, and development flexibility. This model considers 15 cost factors categorized into four areas: product, platform, staff, and project.

3. **COCOMO Advanced:** The Advanced model further refines estimates by factoring in multi-site development, software reuse, and software stability. This version considers additional elements such as the degree of software reuse, the complexity of reliability requirements, and the effects of geographically dispersed teams.

There are three modes of development.

Development mode	Size	Innovation	Deadline	Dev. Environment
Organic	Small	little	not light	Stable
Semi-detached	Medium	Medium	Medium	Medium
Embedded	Large	Greater	Tight	Complex hardware

Table 6.1: Modes of development

Here are the coefficients related to development mode for the intermediate model.

Development mode	a	b	c	d
Organic	2.8	1.05	2.5	0.38

Table 6.2: Coefficients related to development modes for intermediate model

6.1.1 Equations:

$$E = a * (KLOC)^b$$

where,
a = 2.8,b = 1.05, for an organic project.
E = Efforts in person month

6.1.2 Organic project:

Project of moderate size and complexity, where teams with teams with mixed experience levels must meet a mixed rigid and less than rigid requirements(project modway between embedded and oraganic types).

Number of People:

Equation for calculation of number of people required for completion of project, using the COCOMO model is: N=E/D
where,
N = Number of people required
E = Efforts in person-month

D = Duration of project in months

Cost of Project:

Equation for calculation of cost of project, using COCOMO model is:

$$C = D * Cp$$

where,
C = Cost of project
D = Duration in months
Cp = Cost incurred per person-month

6.1.3 Calculation

Efforts:

$$E = a * (KLOC)^b$$

$$E = 2.4 * (3)^{1.05}$$

E = 7.58 person-months
Total of 7.58 person-months are required to complete the project successfully.

Duration of Project:

$$D = c * (E)^d$$

$$D = 2.5 * (7.58)^{0.38}$$

D = 5.39 months
The approximate duration of project is 5 months.

Number of people required for the project:

N=7.58/5
N=1.6
N=2 people Therefore 2 people are required to successfully complete the project on schedule.

Cost of Project:

C=5∗7415 = 37,075 /- Therefore, the cost of the project is 37,075/- (approx)

6.2 Project Schedule and Team Structure

All our project tasks are divided as shown in the table

Task No.	Task Title
T1	Topic Finalization
T2	Requirement specification
T3	Technology Familiarization
T4	System Set up
T5	Concept Review Study
T6	Study of technologies used in the project
T7	Design of user interface
T8	Designing of Constraints Rules
T9	Creation of Connectivity for Services
T10	Creation of database files and rules
T11	Designing the Architectural layout
T12	Creating module using database and rules
T13	Integration of T11 to T7 Tasks
T14	Testing
T15	Documentation Preparation
T16	Maintenance

Table 6.3: Lists Of Tasks

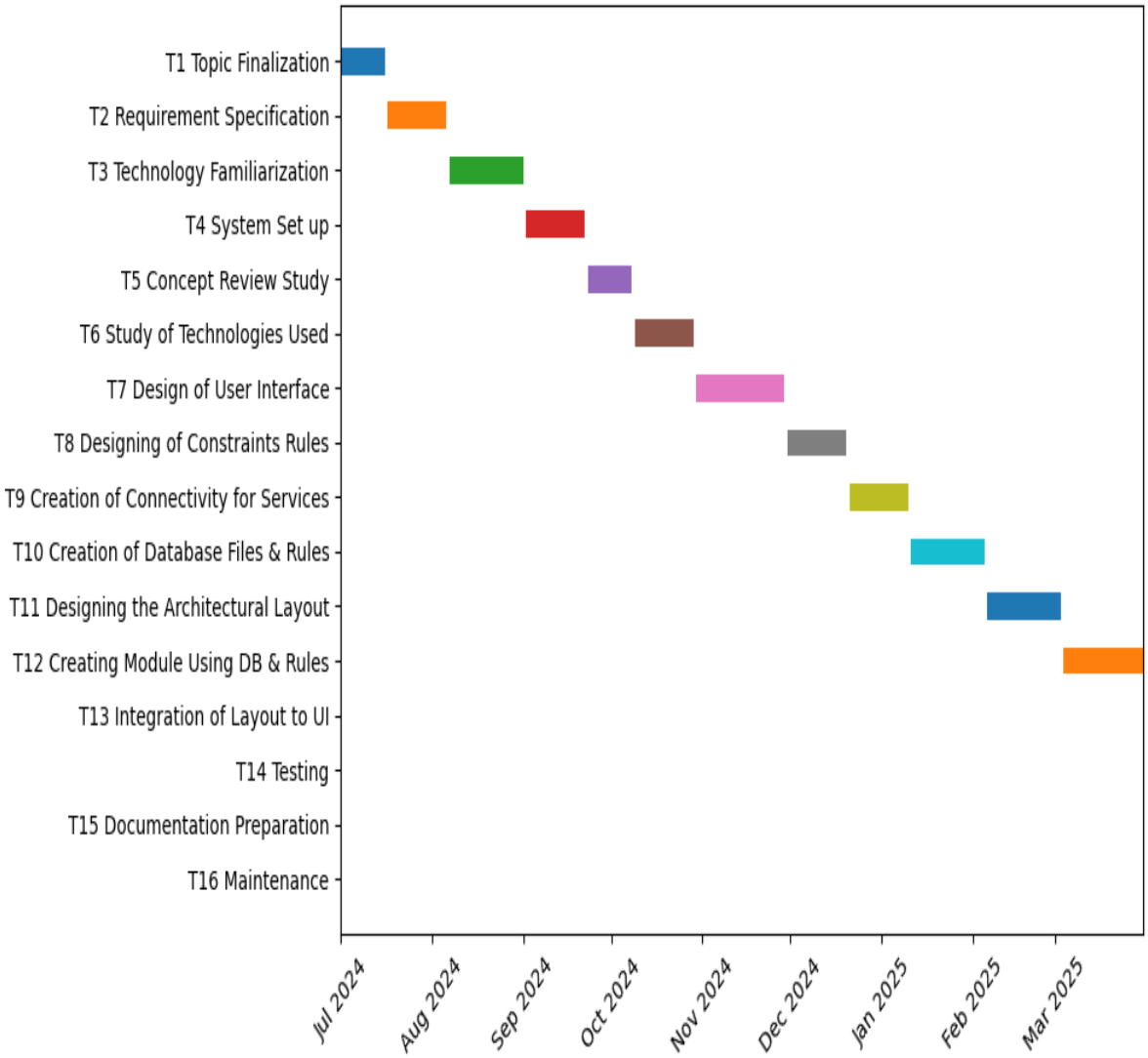


Figure 6.1: Timeline Diagram

Each task is assigned to one or more team members as shown in fig:

Developer ID	Developer Name
D1	Ayesha Shaikh
D2	Pooja Gupta
D3	Shiba Shaikh
D4	Anushka Jadhav

Table 6.4: Lists of Developers

Chapter 7

Software Implementation

7.1 Introduction

This chapter describes the implementation phase of the AI-powered desktop-based presentation tool that integrates gesture recognition and voice command functionalities. The purpose of this project is to eliminate dependency on hardware like remotes, keyboards, and mice for controlling presentations. Users can seamlessly navigate slides through hand gestures and voice commands, making the process contactless, convenient, and accessible. The software uses computer vision to recognize gestures in real time, combined with voice input processing to execute slide actions. Additionally, it supports real-time speech-to-text subtitle generation and multi-language translation, making it an ideal solution for multilingual presentations, especially in classrooms or conferences. The application offers a responsive and intuitive user interface built using Tkinter and leverages powerful Python libraries for backend logic.
The following features are implemented in the system:

- Gesture-based slide control using live webcam feed
- Voice-based commands to control the PowerPoint
- Real-time subtitle generation from speech
- Subtitle translation using Google Translate API
- Minimalist, lightweight GUI for desktop users
- Offline functionality for gesture control (no internet needed)

These features collectively allow the system to function as a complete smart presentation assistant powered by artificial intelligence.

7.2 Databases

The system's database is implemented using MySQL, a reliable relational database that stores user profiles, custom gesture definitions, and presentation preferences in a structured, schema-based format. Each table—such as users and gestures—holds clearly defined columns (e.g. id, username, password in users; id, action, gesturedata, imagepath in gestures) and enforces relationships via primary and foreign keys. This rigid schema ensures data integrity and makes it easy to query for a user's personalized gesture mappings or retrieve user settings at runtime.

MySQL's strong support for transactions, indexing, and optimized joins allows the tool to quickly fetch and update gesture-action mappings during live presentations, ensuring minimal latency. Its mature ecosystem and widespread adoption also provide a wealth of management tools, backup strategies, and monitoring solutions—key for maintaining high availability and performance. Overall, using MySQL gives the project a dependable, performant, and well-understood foundation for storing and retrieving the structured data that drives the AI Presentation Tool.

7.3 Important module and algorithms

7.3.1 Modules

1. **Gesture Recognition Module:**

   The Gesture Recognition Module is responsible for detecting hand gestures from the live webcam feed by leveraging OpenCV for frame capture and Google's MediaPipe for precise landmark detection. MediaPipe identifies key points on the user's hand—such as the tips of the index finger and thumb—and calculates distances and orientations between these landmarks to recognize gestures like "swipe right," "swipe left," "open palm," or "fist." Once a gesture is identified, it is mapped to a corresponding action (for example, advancing to the next slide or returning to the previous one), and PyAutoGUI simulates the appropriate keyboard shortcut in the background. In summary, the algorithm captures each video frame, detects hand landmarks, determines finger positions (up or down), matches the observed pattern to a known gesture, and triggers the associated action via PyAutoGUI.

2. **Voice Command Module:**

   The Voice Command Module enables presenters to control the slideshow through spoken instructions such as "next," "previous," or "start presentation." It begins by recording audio from the microphone, then uses a speech recognition library to transcribe the audio into plain text. The module compares the transcribed text against a set of predefined command keywords, and if a match is found, PyAutoGUI

executes the corresponding action by simulating the relevant keyboard shortcut. The core algorithmic flow involves recording the user's voice, converting speech to text, matching the result against known commands, and executing the mapped action seamlessly.

3. **Real-Time Subtitles Module:**

   To improve accessibility, the Real-Time Subtitles Module captures the presenter's spoken words and displays them as live captions within the application. Using the Vosk speech recognition engine, the module continuously listens to microphone input, transcribes spoken language into text, and updates the GUI—built with Tkinter labels—to show the transcribed captions. Users have the option to enable or disable this feature according to their preference. The simplified algorithmic sequence is: record audio input, transcribe using Vosk, and render the resulting text in the interface in real time.

4. **Subtitle Translation Module:**

   Building on the basic subtitle feature, the Subtitle Translation Module converts live captions into a target language chosen by the user. After the Real-Time Subtitles Module generates the original text, this module sends it to the DeepTranslator library to produce translated text. The translated subtitles are then displayed in the application window alongside—or in place of—the original captions. The algorithmic steps are straightforward: receive the generated subtitle text, translate it via the external API, and update the GUI to show the translated output.

5. **GUI Interaction Module:**

   The GUI Interaction Module, implemented with Tkinter (and optionally CustomTkinter), provides an intuitive control panel for the presenter. Through this interface, users can start or stop gesture recognition, enable or disable real-time subtitles, select their preferred translation language, and monitor live feedback on recognized gestures or voice commands. Buttons, toggle switches, and status indicators ensure that even non-technical users can operate the tool effortlessly. This module ties together all the underlying AI components, making the system both accessible and highly functional.

7.4 Business logic

The business logic of the AI Presentation Tool revolves around ensuring smooth and efficient interaction during presentations by leveraging gesture and voice command recognition. Initially, the system relies on predefined gestures and voice commands that are specifically trained to be recognized, such as "Next Slide," "Pause,"

"Previous Slide," and "End Presentation." Only valid gestures and recognized voice commands are processed, and any invalid inputs are ignored to avoid unintended actions. The tool continuously monitors the input source, which could be either gestures or voice commands, ensuring that only valid actions are accepted based on a predefined set of rules and models. Once a valid gesture or voice command is recognized, the system processes the command and executes the corresponding action, such as advancing to the next slide or pausing the presentation. Real-time feedback is provided to the presenter through visual or auditory cues, ensuring transparency and confirming the executed actions. Additionally, the system remains in a continuous state of readiness to respond to new gestures or voice commands, allowing for uninterrupted interaction throughout the presentation. When the presenter signals the end of the session, the system will promptly conclude the presentation, stop monitoring, and either close the application or return to an idle state.

This business logic ensures a highly interactive, hands-free experience that enhances user engagement, improves the flow of the presentation, and provides professionals and educators with an effective tool to focus on content delivery rather than manual controls.

Chapter 8

Software Testing

8.1 Introduction

We have started software testing started as early in the software development process as possible, and it is integrated into the process of determining requirements. One stage of a lifecycle is testing. The lifecycle of software development is one in which we identify a requirement, write some code to address it, and then assess if we have satisfied the stakeholders, including the users, owners, and other parties with an interest in the product's functionality.

8.1.1 Test cases for Gesture Control Module

No	**Behaviour Description**	**Property**
1	Unique test case ID	GC-01
2	Test case name	Detect gesture for next slide
3	Prerequisites	System setup completed with module installed
4	Test case description	Detect swipe right gesture for next slide
5	Input	Perform a right swipe gesture
6	Expected results	Slide moves to the next
7	Actual results	Slide moves to the next
8	Pass/fail	Pass

Table 8.1: Test Case: Detect gesture for next slide (GC-01)

No	Behaviour Description	Property
1	Unique test case ID	GC-02
2	Test case name	Detect gesture for previous slide
3	Prerequisites	System setup completed with module installed
4	Test case description	Detect swipe left gesture for previous slide
5	Input	Perform a left swipe gesture
6	Expected results	Slide moves to the previous
7	Actual results	Slide moves to the previous
8	Pass/fail	Pass

Table 8.2: Test Case: Detect gesture for previous slide (GC-02)

No	Behaviour Description	Property
1	Unique test case ID	GC-03
2	Test case name	Detect gesture to start presentation
3	Prerequisites	System setup completed with module installed
4	Test case description	Detect thumb-up gesture to start presentation
5	Input	Show thumb-up gesture
6	Expected results	Presentation starts
7	Actual results	Presentation starts
8	Pass/fail	Pass

Table 8.3: Test Case: Detect gesture to start presentation (GC-03)

No	Behaviour Description	Property
1	Unique test case ID	GC-04
2	Test case name	Detect gesture to end presentation
3	Prerequisites	System setup completed with module installed
4	Test case description	Detect thumb-down gesture to end presentation
5	Input	Show thumb-down gesture
6	Expected results	Presentation ends
7	Actual results	Presentation ends
8	Pass/fail	Pass

Table 8.4: Test Case: Detect gesture to end presentation (GC-04)

No	Behaviour Description	Property
1	Unique test case ID	GC-05
2	Test case name	Ignore unrecognized gestures
3	Prerequisites	System setup completed with module installed
4	Test case description	Ignore unrecognized gestures
5	Input	Perform a random hand movement
6	Expected results	No action is triggered
7	Actual results	No action is triggered
8	Pass/fail	Pass

Table 8.5: Test Case: Ignore unrecognized gestures (GC-05)

8.1.2 Test cases for Voice Control Module

No	Behaviour Description	Property
1	Unique test case ID	VC-01
2	Test case name	Lock gestures
3	Prerequisites	System setup completed with module installed
4	Test case description	Lock gestures using voice command
5	Input	Say "Lock gestures"
6	Expected results	Gestures are locked and status message displayed
7	Actual results	Gestures are locked and status message displayed
8	Pass/fail	Pass

Table 8.6: Test Case: Lock gestures (VC-01)

No	Behaviour Description	Property
1	Unique test case ID	VC-02
2	Test case name	Resume gestures
3	Prerequisites	System setup completed with module installed
4	Test case description	Resume gestures using voice command
5	Input	Say " Resume gestures"
6	Expected results	Gestures are unlocked and status message displayed
7	Actual results	Gestures are not unlocked and status message displayed
8	Pass/fail	Fail

Table 8.7: Test Case: Resume gestures (VC-02)

No	Behaviour Description	Property
1	Unique test case ID	VC-03
2	Test case name	Ignore unrecognized voice commands
3	Prerequisites	System setup completed with module installed
4	Test case description	Ignore unrecognized voice commands
5	Input	Say an unrelated word (e.g., "Hello")
6	Expected results	No action is triggered
7	Actual results	No action is triggered
8	Pass/fail	Pass

Table 8.8: Test Case: Ignore unrecognized voice commands (VC-03)

No	Behaviour Description	Property
1	Unique test case ID	VC-04
2	Test case name	Recognize 'next' command and navigate
3	Prerequisites	System setup completed with module installed
4	Test case description	Recognize 'next' command and navigate
5	Input	Say "Next slide"
6	Expected results	Moves to next slide
7	Actual results	Moves to next slide
8	Pass/fail	Pass

Table 8.9: Test Case: Recognize 'next' command and navigate (VC-04)

No	Behaviour Description	Property
1	Unique test case ID	VC-05
2	Test case name	Recognize 'previous' command and navigate
3	Prerequisites	System setup completed with module installed
4	Test case description	Recognize 'previous' command and navigate
5	Input	Say "Previous slide"
6	Expected results	Moves to previous slide
7	Actual results	Moves to previous slide
8	Pass/fail	Pass

Table 8.10: Test Case: Recognize 'previous' command and navigate (VC-05)

No	Behaviour Description	Property
1	Unique test case ID	VC-06
2	Test case name	Recognize 'go to slide [number]' command
3	Prerequisites	System setup completed with module installed
4	Test case description	Recognize 'go to slide [number]' command
5	Input	Say "Go to slide 5"
6	Expected results	Navigates to slide 5
7	Actual results	Navigates to slide 5
8	Pass/fail	Pass

Table 8.11: Test Case: Recognize 'go to slide [number]' command (VC-06)

No	Behaviour Description	Property
1	Unique test case ID	VC-07
2	Test case name	Recognize 'start presentation' command
3	Prerequisites	System setup completed with module installed
4	Test case description	Recognize 'start presentation' command
5	Input	Say "Start presentation"
6	Expected results	Starts slideshow
7	Actual results	Starts slideshow
8	Pass/fail	Pass

Table 8.12: Test Case: Recognize 'start presentation' command (VC-07)

No	Behaviour Description	Property
1	Unique test case ID	VC-08
2	Test case name	Recognize 'stop presentation' command
3	Prerequisites	System setup completed with module installed
4	Test case description	Recognize 'stop presentation' command
5	Input	Say "Stop presentation"
6	Expected results	Exits slideshow
7	Actual results	Exits slideshow
8	Pass/fail	Pass

Table 8.13: Test Case: Recognize 'stop presentation' command (VC-08)

No	Behaviour Description	Property
1	Unique test case ID	VC-09
2	Test case name	Ignore unrecognized commands
3	Prerequisites	System setup completed with module installed
4	Test case description	Ignore unrecognized commands
5	Input	Say an unrelated word (e.g., "Hello")
6	Expected results	No action triggered
7	Actual results	No action triggered
8	Pass/fail	Pass

Table 8.14: Test Case: Ignore unrecognized commands (VC-09)

No	Behaviour Description	Property
1	Unique test case ID	VC-10
2	Test case name	Handle incorrect slide number input
3	Prerequisites	System setup completed with module installed
4	Test case description	Handle incorrect slide number input
5	Input	Say "Go to slide 999" (invalid slide)
6	Expected results	No navigation occurs
7	Actual results	No navigation occurs
8	Pass/fail	Pass

Table 8.15: Test Case: Handle incorrect slide number input (VC-10)

8.1.3 Test cases for Real-time Subtitles Module

No	Behaviour Description	Property
1	Unique test case ID	RTS-01
2	Test case name	Recognize and display spoken words
3	Prerequisites	System setup completed with module installed
4	Test case description	Recognize and display spoken words
5	Input	Speak clearly into the microphone
6	Expected results	Spoken words appear as subtitles
7	Actual results	Spoken words appear as subtitles
8	Pass/fail	Pass

Table 8.16: Test Case: Recognize and display spoken words (RTS-01)

No	Behaviour Description	Property
1	Unique test case ID	RTS-02
2	Test case name	Handle no speech input
3	Prerequisites	System setup completed with module installed
4	Test case description	Handle no speech input
5	Input	Stay silent for 10 seconds
6	Expected results	Subtitle overlay remains empty
7	Actual results	Subtitle overlay remains empty
8	Pass/fail	Pass

Table 8.17: Test Case: Handle no speech input (RTS-02)

No	Behaviour Description	Property
1	Unique test case ID	RTS-03
2	Test case name	Start subtitles overlay
3	Prerequisites	System setup completed with module installed
4	Test case description	Start subtitles overlay
5	Input	Call start_subtitle() function
6	Expected results	Subtitle overlay appears on screen
7	Actual results	Subtitle overlay appears on screen
8	Pass/fail	Pass

Table 8.18: Test Case: Start subtitles overlay (RTS-03)

No	Behaviour Description	Property
1	Unique test case ID	RTS-04
2	Test case name	Stop subtitles overlay
3	Prerequisites	System setup completed with module installed
4	Test case description	Stop subtitles overlay
5	Input	Call stop_subtitle() function
6	Expected results	Subtitle overlay disappears
7	Actual results	Subtitle overlay disappears
8	Pass/fail	Pass

Table 8.19: Test Case: Stop subtitles overlay (RTS-04)

8.1.4 Test cases for Real-time Translation Module

No	Behaviour Description	Property
1	Unique test case ID	RTT-01
2	Test case name	Start real-time translation
3	Prerequisites	System setup completed with module installed
4	Test case description	Start real-time translation
5	Input	Call start_real_time_translation() function
6	Expected results	Translation overlay appears on screen
7	Actual results	Translation overlay appears on screen
8	Pass/fail	Pass

Table 8.20: Test Case: Start real-time translation (RTT-01)

No	Behaviour Description	Property
1	Unique test case ID	RTT-02
2	Test case name	Stop translation overlay
3	Prerequisites	System setup completed with module installed
4	Test case description	Stop translation overlay
5	Input	Call stop_real_time_translation() function
6	Expected results	Translation overlay disappears
7	Actual results	Translation overlay disappears
8	Pass/fail	Pass

Table 8.21: Test Case: Stop translation overlay (RTT-02)

No	Behaviour Description	Property
1	Unique test case ID	RTT-03
2	Test case name	Verify English to Hindi translation
3	Prerequisites	System setup completed with module installed
4	Test case description	Verify English to Hindi translation
5	Input	Select 'English to Hindi' option then Speak in English
6	Expected results	Translated text appears in Hindi
7	Actual results	Translated text appears in Hindi
8	Pass/fail	Pass

Table 8.22: Test Case: Verify English to Hindi translation (RTT-03)

No	Behaviour Description	Property
1	Unique test case ID	RTT-04
2	Test case name	Verify Hindi to English translation
3	Prerequisites	System setup completed with module installed
4	Test case description	Verify Hindi to English translation
5	Input	Select 'Hindi to English' option then Speak in Hindi
6	Expected results	Translated text appears in English
7	Actual results	Translated text appears in English
8	Pass/fail	Pass

Table 8.23: Test Case: Verify Hindi to English translation (RTT-04)

No	Behaviour Description	Property
1	Unique test case ID	RTT-05
2	Test case name	Verify English to French translation
3	Prerequisites	System setup completed with module installed
4	Test case description	Verify English to French translation
5	Input	Select 'English to French' option then Speak in English
6	Expected results	Translated text appears in French
7	Actual results	Translated text appears in French
8	Pass/fail	Pass

Table 8.24: Test Case: Verify English to French translation (RTT-05)

No	Behaviour Description	Property
1	Unique test case ID	RTT-06
2	Test case name	Verify English to German translation
3	Prerequisites	System setup completed with module installed
4	Test case description	Verify English to German translation
5	Input	Select 'English to German' option then Speak in English
6	Expected results	Translated text appears in German
7	Actual results	Translated text appears in German
8	Pass/fail	Pass

Table 8.25: Test Case: Verify English to German translation (RTT-06)

No	Behaviour Description	Property
1	Unique test case ID	RTT-07
2	Test case name	Handle non-supported language input
3	Prerequisites	System setup completed with module installed
4	Test case description	Handle non-supported language input
5	Input	Speak in an unsupported language
6	Expected results	Displays 'Could not understand Audio'
7	Actual results	Displays 'Could not understand Audio'
8	Pass/fail	Pass

Table 8.26: Test Case: Handle non-supported language input (RTT-07)

8.1.5 Test cases for Customized Gesture Module

No	Behaviour Description	Property
1	Unique test case ID	CG-01
2	Test case name	Store a new custom gesture
3	Prerequisites	System setup completed with module installed
4	Test case description	Store a new custom gesture in database
5	Input	Perform a new gesture then Assign an action and save
6	Expected results	Gesture is saved in the database with angles and image
7	Actual results	Gesture is saved in the database with angles and image
8	Pass/fail	Pass

Table 8.27: Test Case: Store a new custom gesture (CG-01)

No	Behaviour Description	Property
1	Unique test case ID	CG-02
2	Test case name	Retrieve stored gestures
3	Prerequisites	System setup completed with module installed
4	Test case description	Retrieve stored gestures from database
5	Input	Navigate to gesture management then View stored gestures
6	Expected results	List of stored gestures is displayed
7	Actual results	List of stored gestures is displayed
8	Pass/fail	Pass

Table 8.28: Test Case: Retrieve stored gestures (CG-02)

No	Behaviour Description	Property
1	Unique test case ID	CG-03
2	Test case name	Detect a saved and execute custom gesture
3	Prerequisites	System setup completed with module installed
4	Test case description	Detect a saved custom gesture and execute action
5	Input	Perform a stored custom gesture
6	Expected results	Corresponding action is triggered
7	Actual results	Corresponding action is triggered
8	Pass/fail	Pass

Table 8.29: Test Case: Detect a saved and execute custom gesture (CG-03)

No	Behaviour Description	Property
1	Unique test case ID	CG-04
2	Test case name	Delete a stored custom gesture
3	Prerequisites	System setup completed with module installed
4	Test case description	Delete a stored custom gesture
5	Input	Navigate to gesture management then Delete a gesture
6	Expected results	Gesture is removed from the database
7	Actual results	Gesture is removed from the database
8	Pass/fail	Pass

Table 8.30: Test Case: Delete a stored custom gesture (CG-04)

No	Behaviour Description	Property
1	Unique test case ID	CG-05
2	Test case name	Update an existing gesture
3	Prerequisites	System setup completed with module installed
4	Test case description	Update an existing gesture
5	Input	Navigate to gesture management then select a gesture and update it
6	Expected results	Gesture angles and image are updated
7	Actual results	Gesture angles and image are updated
8	Pass/fail	Pass

Table 8.31: Test Case: Update an existing gesture (CG-05)

No	Behaviour Description	Property
1	Unique test case ID	CG-06
2	Test case name	Handle duplicate gestures
3	Prerequisites	System setup completed with module installed
4	Test case description	Handle duplicate gestures for different actions
5	Input	Try saving two different actions with similar gestures
6	Expected results	Error message is displayed
7	Actual results	Latest updated gesture is executed
8	Pass/fail	Fail

Table 8.32: Test Case: Handle duplicate gestures (CG-06)

Chapter 9

Result

9.1 Snapshots of the results

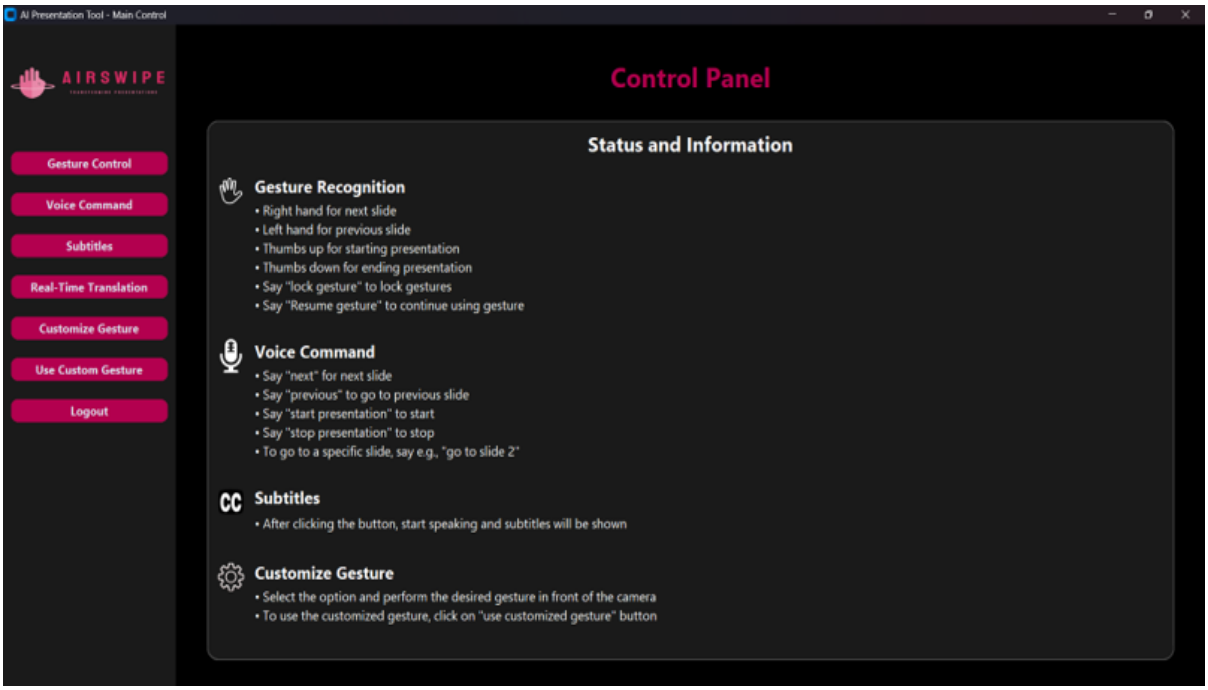


Figure 9.1: Control Panel

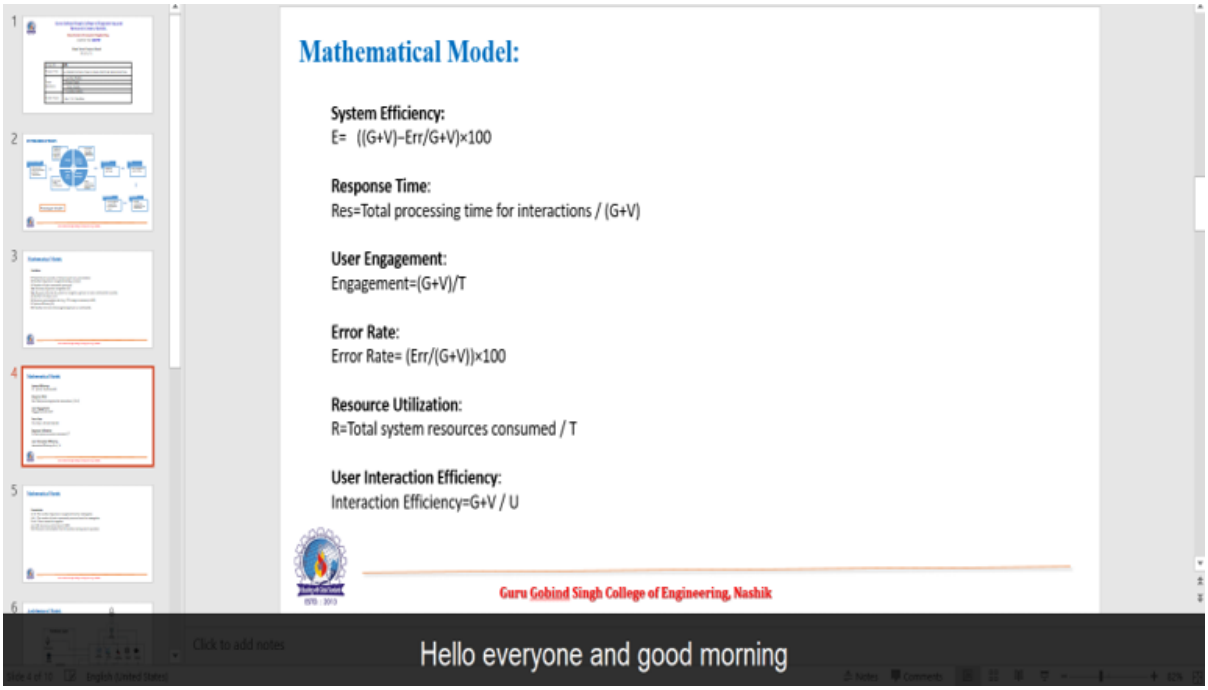


Figure 9.2: Real-Time subtitles

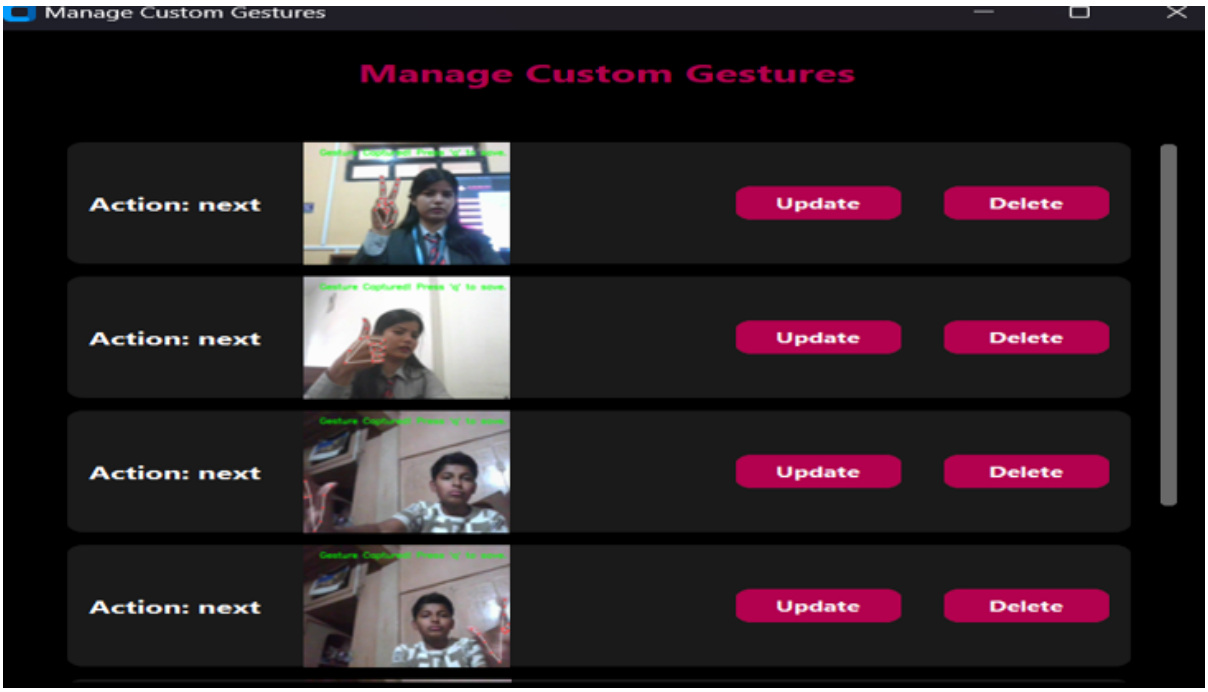


Figure 9.3: Manage Custom Gestures

Figure 9.4: Select Custom Gesture

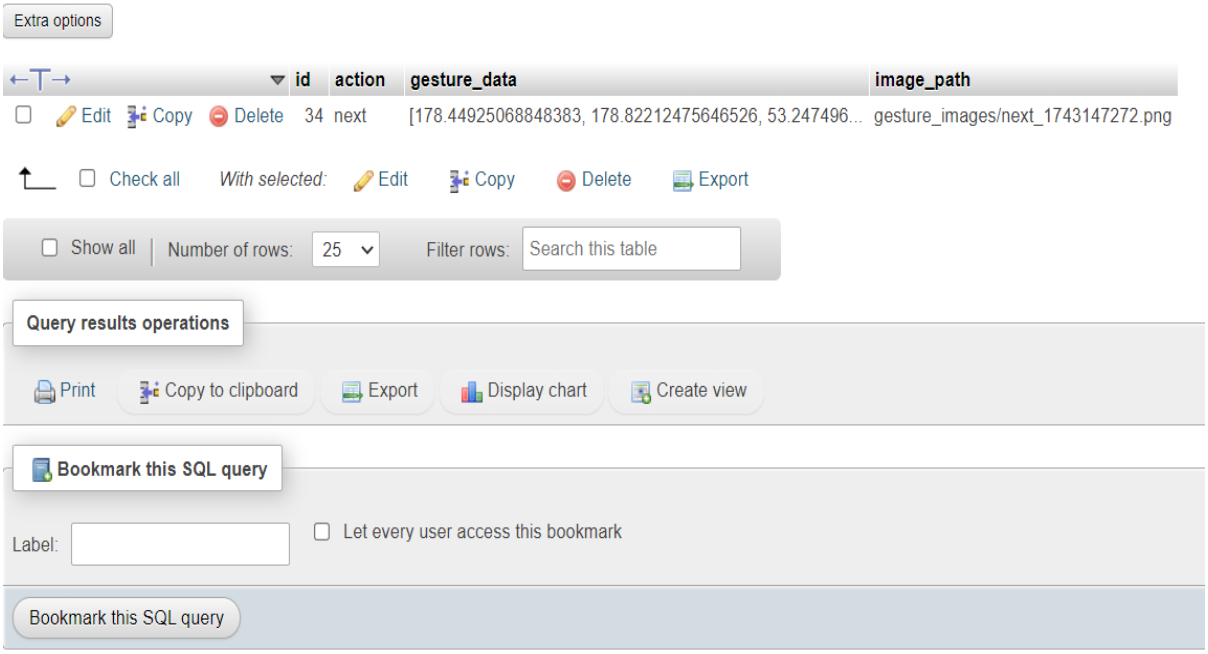


Figure 9.5: Storing Custom Gestures

9.2 Result

Time taken by each functionality of the system:

Parameter	Description	Measured Time
GRT (Gesture Recognition Time)	Time taken to recognize and process a gesture input.	0.2s
VRT (Voice Recognition Time)	Response time for processing voice commands and executing corresponding actions.	2.0s
RST (Real-time Subtitles Processing)	Delay in generating subtitles from spoken input.	1.0s
RTT (Real-time Translation Latency)	Time taken to translate speech into another language.	2.0s
UILT (User Interface Load Time)	Time required for the application dashboard to fully load.	0.8s

Table 9.1: Performance Parameters and Measured Times

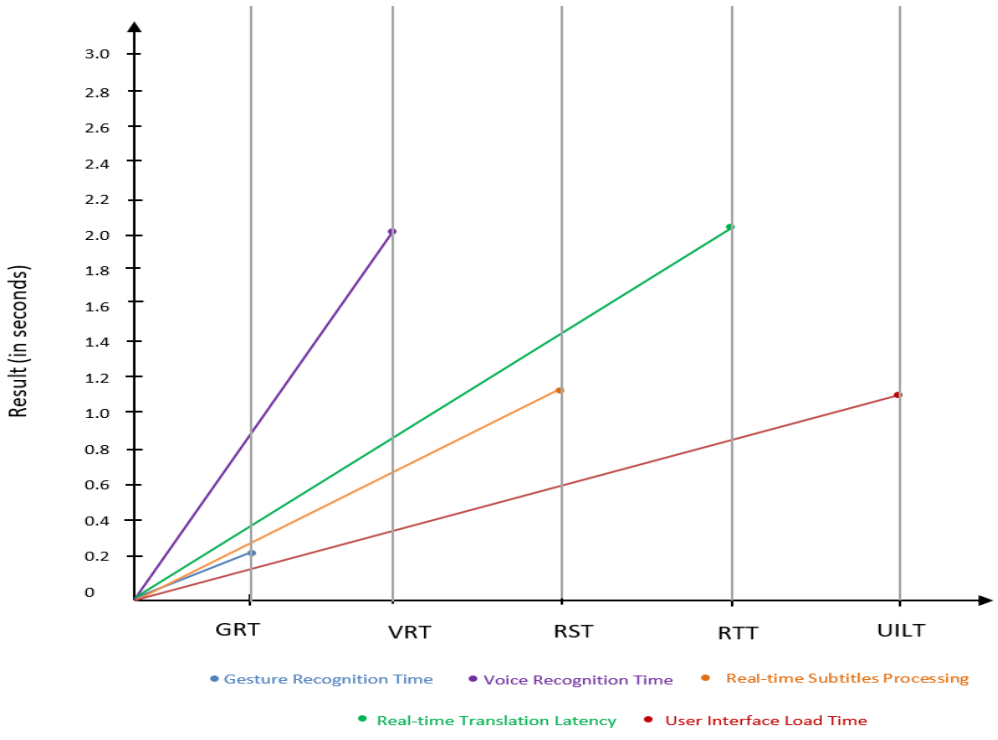


Figure 9.6: Speed Visualization

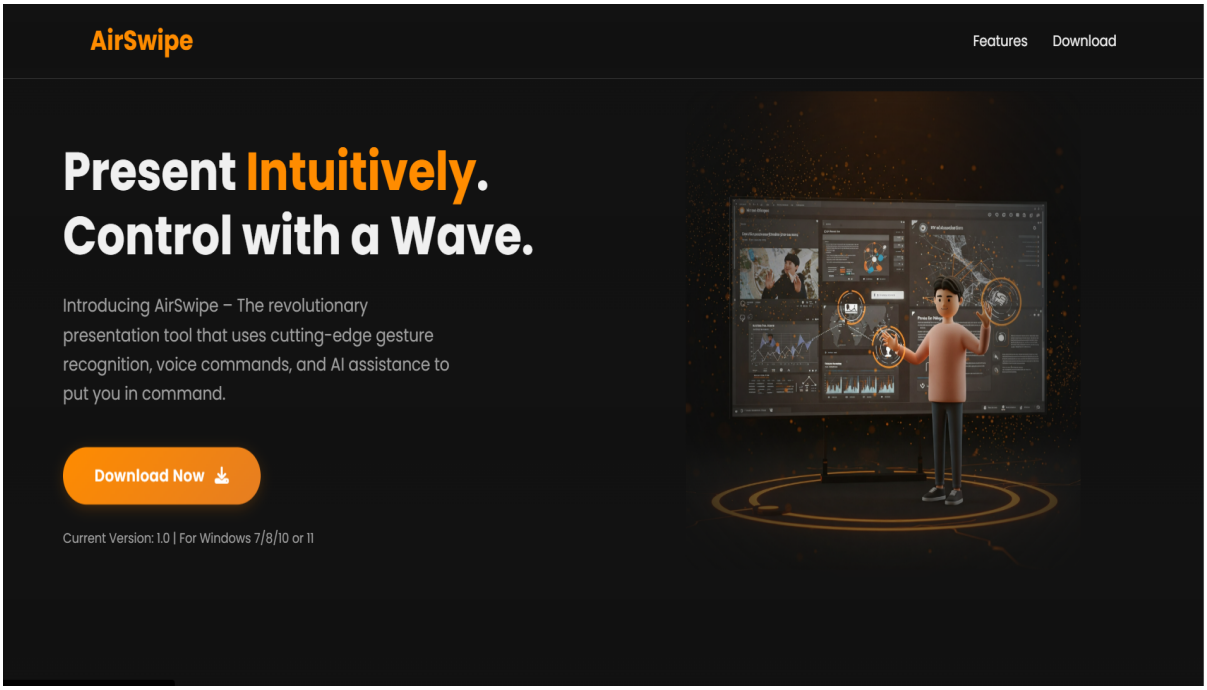


Figure 9.7: Website UI

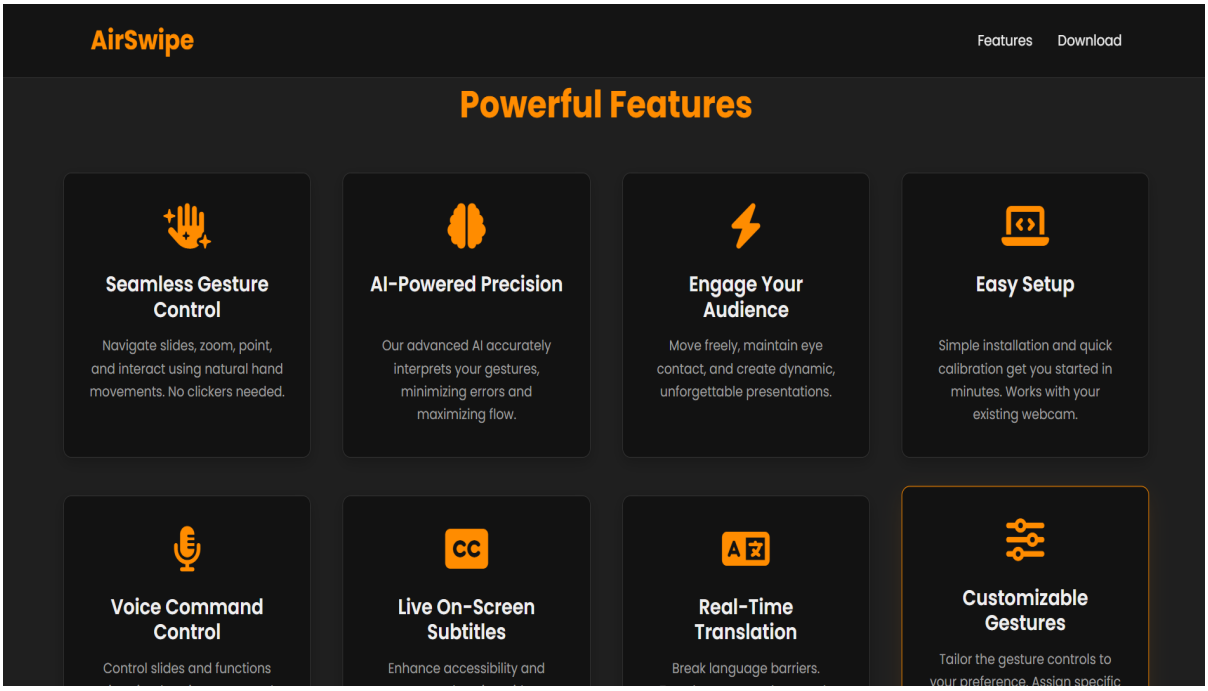


Figure 9.8: Website UI

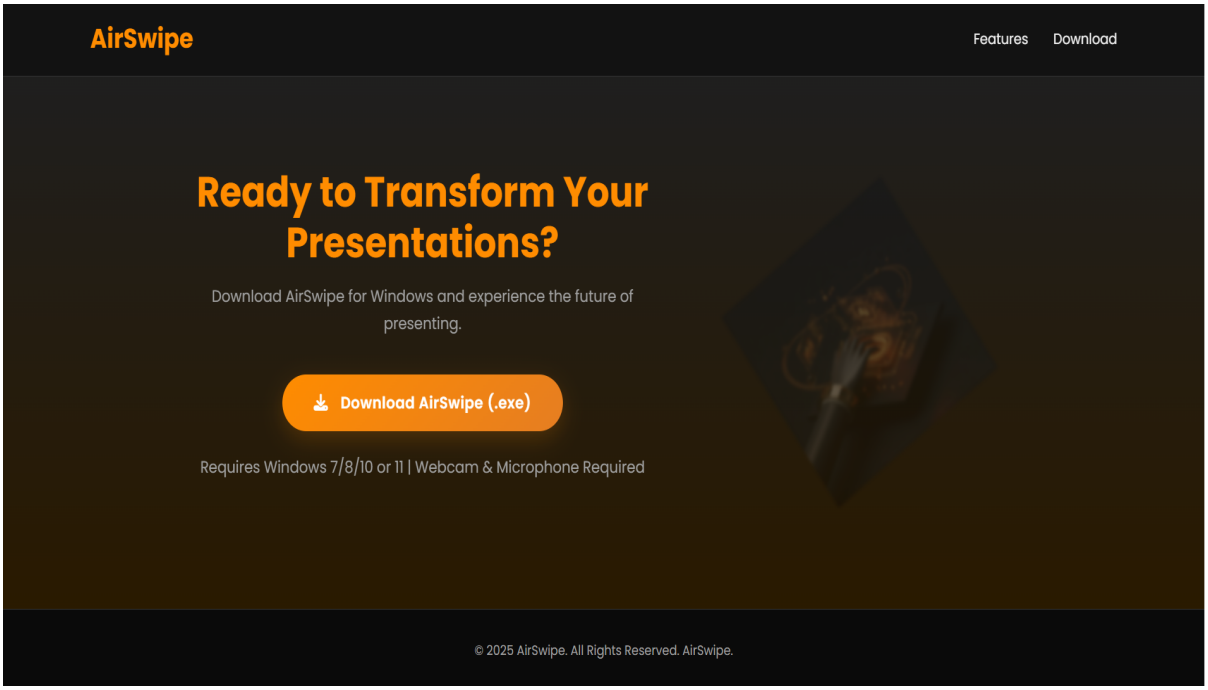


Figure 9.9: Website UI

Chapter 10

Deployment and Maintenance

10.1 Deployment and Maintenance

10.1.1 Installation and un-installation

For the deployment of the **AirSwipe: An AI-Driven Presentation Controller**, we created a standalone Windows executable (.exe) using PyInstaller. This allowed us to run the application on systems without requiring a separate Python installation or manual setup of dependencies.

Steps Followed in Deployment:

1. Preparing the Environment

First, we ensured all required Python libraries—such as `MediaPipe`, `OpenCV`, `SpeechRecognition`, `Tkinter`, and other dependencies were properly installed and tested in a virtual environment to avoid version conflicts.

2. Creating the Executable with PyInstaller

We used PyInstaller to convert our main Python script into a `.exe` file. The command we used was:

```
pyinstaller --onefile --noconsole welcome_window.py
```

`--onefile`: Packages the entire application into a single `.exe` file.

`--noconsole`: Hides the terminal window for a clean GUI-based user experience.

The generated `.exe` was found in the `/dist` directory.

3. Handling External Files

Since our project also uses model files , image files, and JSON configurations, we ensured these were included either by using the `--add-data` flag in PyInstaller or placing them in known paths that the app could locate at runtime.

4. Creating the Installer

After the `.exe` was successfully generated, we used **Inno Setup** to create an installer `.exe`. This installer:

- Placed the main `welcomewindow.exe` in the Program Files directory.
- Created a desktop shortcut.
- Copied necessary config/model files to the application folder.
- Optionally created a user-specific folder for local database storage.

5. Testing the Installer

We tested the final installer on a clean Windows machine (without Python installed) to verify that:

- The installation completes successfully.
- All gestures and voice commands work correctly.
- All necessary resources are bundled.
- The software launches properly with GUI.

6. Uninstallation

An uninstaller was included as part of the Inno Setup script. The app can be removed via the “Add or Remove Programs” feature in Windows. Uninstallation deletes all installed files and shortcuts.

We have deployed our website ”https://airswipe.netlify.app/” from which users can download the .exe file and use our software into their system.

10.1.2 Maintenance

Software maintenance is an essential process that includes the modification of a software product after it has been deployed in order to correct faults, improve performance, or introduce new features. Maintenance typically involves:

- **Bug Fixes**: After the initial deployment, if any bugs or issues are identified by users, these need to be addressed by modifying the code. The `.exe` file will then need to be recompiled and distributed to users.
- **Updating Dependencies**: Over time, the libraries or frameworks used in the project may be updated. In such cases, the relevant dependencies will need to be updated in the code, and a new version of the `.exe` file will be generated.
- **Performance Optimization**: As new hardware becomes available or as user feedback is gathered, optimizing the performance of the AI models or gesture recognition algorithms might be necessary. This will also require updating the software and redistributing the new version.
- **Security Patches**: Security vulnerabilities may arise over time due to outdated libraries or external threats. In such cases, the software must be updated with security patches to protect users' data and privacy.
- **Feature Updates**: New features, such as additional gesture recognition or support for new presentation software, may be added in response to user demands. After developing these features, the new `.exe` file must be built and updated.
- **Backup and Restore**: In case of software updates, a backup of user preferences and settings might be required to ensure that users don't lose important data when updating to the new version.

After any updates or bug fixes, the new `.exe` file must be redistributed to users. Users can download the new version via the application's website, or you can integrate an auto-update feature that notifies users when an update is available.

Conclusion and Future Scope

Conclusion

This system aims to revolutionize the way users interact with presentation tools. By integrating gesture recognition and voice commands, users can control slides naturally and efficiently. Additionally, real-time subtitle generation and speech translation makes the system highly inclusive, allowing it to be used in educational seminars, business meetings, and multilingual conferences. Through this hands-free and AI-based interaction, the tool enhances accessibility, reduces dependency on hardware tools like remotes or clickers, and brings innovation to the presentation experience.

Future Scope

Looking ahead, the AI-powered presentation tool using gesture recognition has great potential for growth. As technology continues to advance, we can expect more accurate and responsive gesture recognition, making the tool even easier and more natural to use. Expanding the range of gestures and incorporating more languages for voice control would help make the tool more accessible to a global audience. Future versions could also include features like real-time audience interaction, where presenters can receive feedback or gauge engagement during their presentation. Additionally, integrating augmented and virtual reality could transform presentations into immersive experiences. With ongoing improvements in AI, the tool could also evolve to provide more personalized features, making every presentation feel unique and tailored to the speaker's style. Ultimately, this tool could change the way presentations are delivered, making them more dynamic and interactive for both presenters and audiences alike.

References

[1] R. M. Shakya, S. P. Sharma, and N. Shrestha, "The Use of Hand Gestures as a Tool for Presentation," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 11, 2023.

[2] S. R. Patel, "Gesture Recognition: Applications in Education and Presentations," *IEEE Transactions on Human-Machine Systems*, vol. 53, no. 3, pp. 250-260, 2023.

[3] M. K. Jha and A. D. Gupta, "A Study on Gesture Recognition in Interactive Presentations," *ITM Conference Proceedings, ICACC 2022*.

[4] A. Lee and B. Chan, "Enhancing Presentations with Gesture Recognition Technologies," *IEEE Access*, vol. 10, pp. 500-510, 2022.

[5] J. P. Brown, "The Future of Interactive Presentations: Gesture and Touch," *IEEE Transactions on Education*, vol. 65, no. 2, pp. 120-130, 2022.

[6] L. Zhang, "Utilizing Kinect for Enhanced Presentation Control," *IEEE Transactions on Multimedia*, vol. 24, pp. 1-10, 2022.

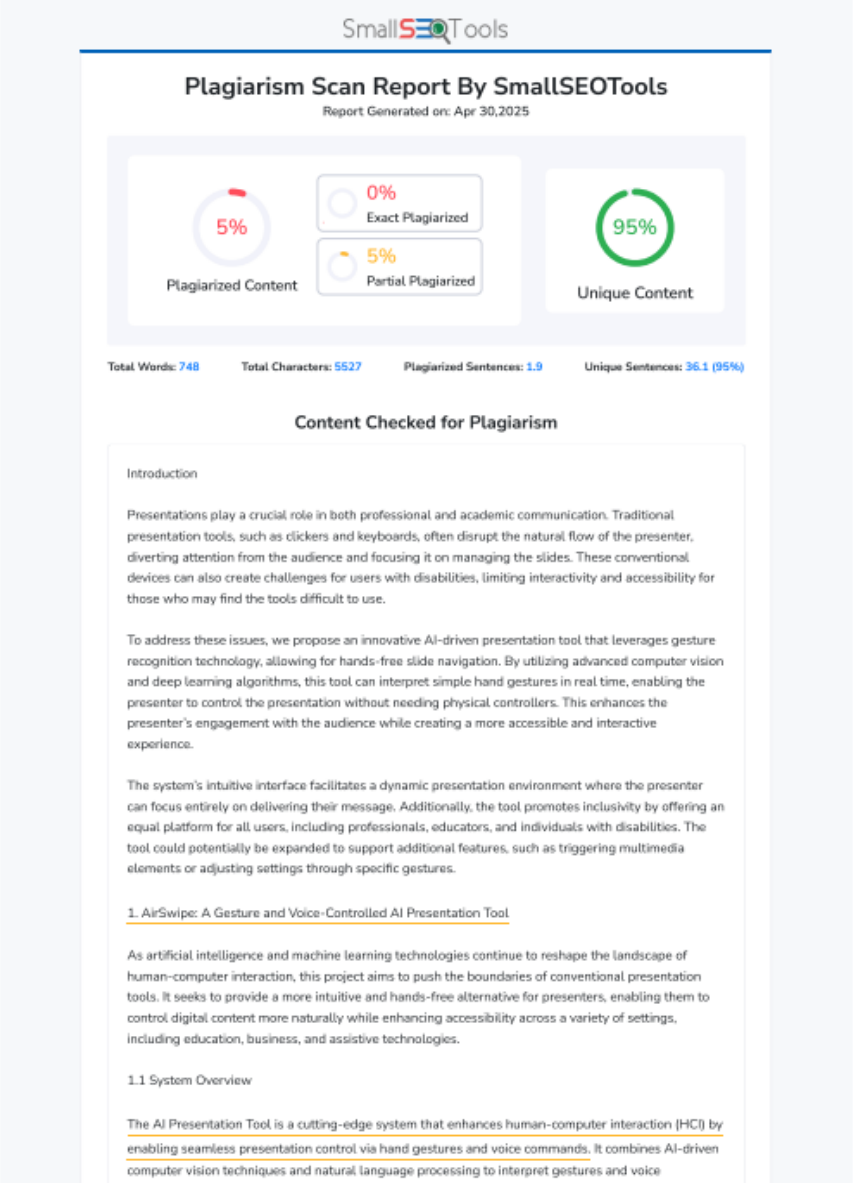
[7] S. Powar, S. Kadam, S. Malage, and P. Shingane, "Automated Digital Presentation Control using Hand Gesture Technique," 2022.

[8] O. Mubin, J. Henderson, and C. Bartneck, "A Comparative Analysis of Real-Time Open-Source Speech Recognition Tools for Social Robots," 2020.

[9] T. Kirishima, K. Sato, and K. Chihara, "Real-Time Gesture Recognition by Learning and Selective Control of Visual Interest Points," March 2005.

[10] College of Computer and Cyber Security, Communication University of China, "Video Subtitle Location and Recognition Based on Edge Features," 2022.

Plagiarism Report



SmallSEOTools

Plagiarism Scan Report By SmallSEOTools

Report Generated on: Apr 30,2025

5% Plagiarized Content

0% Exact Plagiarized

5% Partial Plagiarized

95% Unique Content

Total Words: 748 Total Characters: 5527 Plagiarized Sentences: 1.9 Unique Sentences: 36.1 (95%)

Content Checked for Plagiarism

Introduction

Presentations play a crucial role in both professional and academic communication. Traditional presentation tools, such as clickers and keyboards, often disrupt the natural flow of the presenter, diverting attention from the audience and focusing it on managing the slides. These conventional devices can also create challenges for users with disabilities, limiting interactivity and accessibility for those who may find the tools difficult to use.

To address these issues, we propose an innovative AI-driven presentation tool that leverages gesture recognition technology, allowing for hands-free slide navigation. By utilizing advanced computer vision and deep learning algorithms, this tool can interpret simple hand gestures in real time, enabling the presenter to control the presentation without needing physical controllers. This enhances the presenter's engagement with the audience while creating a more accessible and interactive experience.

The system's intuitive interface facilitates a dynamic presentation environment where the presenter can focus entirely on delivering their message. Additionally, the tool promotes inclusivity by offering an equal platform for all users, including professionals, educators, and individuals with disabilities. The tool could potentially be expanded to support additional features, such as triggering multimedia elements or adjusting settings through specific gestures.

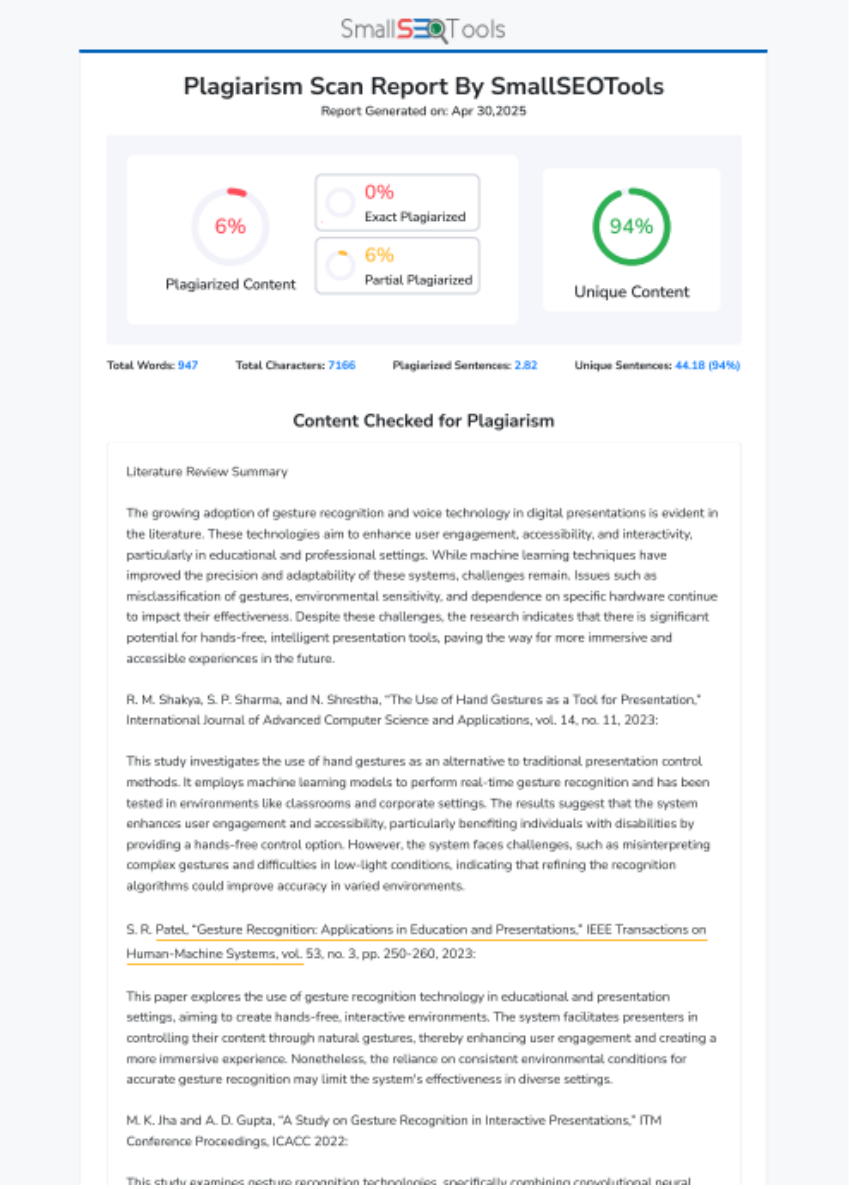
1. AirSwipe: A Gesture and Voice-Controlled AI Presentation Tool

As artificial intelligence and machine learning technologies continue to reshape the landscape of human-computer interaction, this project aims to push the boundaries of conventional presentation tools. It seeks to provide a more intuitive and hands-free alternative for presenters, enabling them to control digital content more naturally while enhancing accessibility across a variety of settings, including education, business, and assistive technologies.

1.1 System Overview

The AI Presentation Tool is a cutting-edge system that enhances human-computer interaction (HCI) by enabling seamless presentation control via hand gestures and voice commands. It combines AI-driven computer vision techniques and natural language processing to interpret gestures and voice

Figure 10.1: Plagiarism Report for Chapter 1



SmallSEOTools

Plagiarism Scan Report By SmallSEOTools

Report Generated on: Apr 30,2025

6% Plagiarized Content

0% Exact Plagiarized

6% Partial Plagiarized

94% Unique Content

Total Words: 947 Total Characters: 7166 Plagiarized Sentences: 2.82 Unique Sentences: 44.18 (94%)

Content Checked for Plagiarism

Literature Review Summary

The growing adoption of gesture recognition and voice technology in digital presentations is evident in the literature. These technologies aim to enhance user engagement, accessibility, and interactivity, particularly in educational and professional settings. While machine learning techniques have improved the precision and adaptability of these systems, challenges remain. Issues such as misclassification of gestures, environmental sensitivity, and dependence on specific hardware continue to impact their effectiveness. Despite these challenges, the research indicates that there is significant potential for hands-free, intelligent presentation tools, paving the way for more immersive and accessible experiences in the future.

R. M. Shakya, S. P. Sharma, and N. Shrestha, "The Use of Hand Gestures as a Tool for Presentation," International Journal of Advanced Computer Science and Applications, vol. 14, no. 11, 2023:

This study investigates the use of hand gestures as an alternative to traditional presentation control methods. It employs machine learning models to perform real-time gesture recognition and has been tested in environments like classrooms and corporate settings. The results suggest that the system enhances user engagement and accessibility, particularly benefiting individuals with disabilities by providing a hands-free control option. However, the system faces challenges, such as misinterpreting complex gestures and difficulties in low-light conditions, indicating that refining the recognition algorithms could improve accuracy in varied environments.

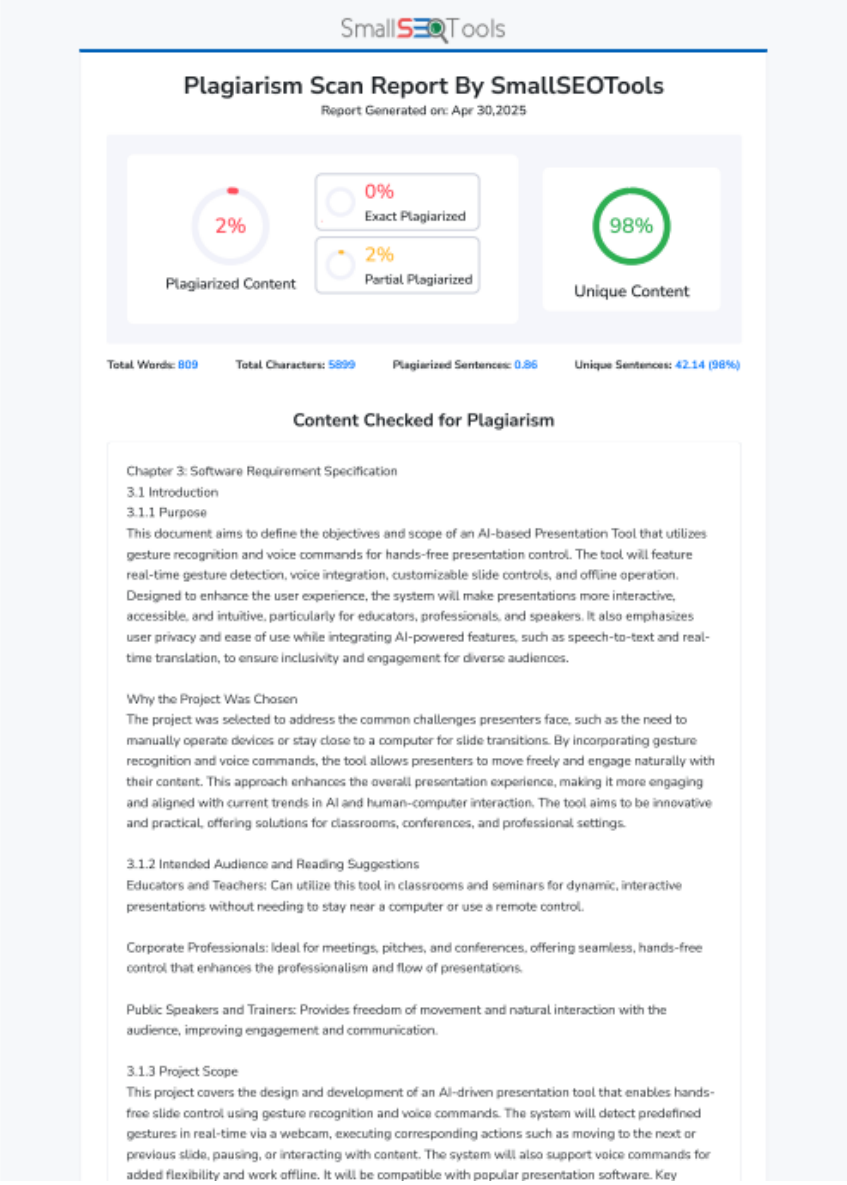
S. R. Patel, "Gesture Recognition: Applications in Education and Presentations," IEEE Transactions on Human-Machine Systems, vol. 53, no. 3, pp. 250-260, 2023:

This paper explores the use of gesture recognition technology in educational and presentation settings, aiming to create hands-free, interactive environments. The system facilitates presenters in controlling their content through natural gestures, thereby enhancing user engagement and creating a more immersive experience. Nonetheless, the reliance on consistent environmental conditions for accurate gesture recognition may limit the system's effectiveness in diverse settings.

M. K. Jha and A. D. Gupta, "A Study on Gesture Recognition in Interactive Presentations," ITM Conference Proceedings, ICACC 2022:

This study examines gesture recognition technologies, specifically combining convolutional neural

Figure 10.2: Plagiarism Report for Chapter 2



SmallSEOTools

Plagiarism Scan Report By SmallSEOTools

Report Generated on: Apr 30,2025

2% Plagiarized Content

0% Exact Plagiarized

2% Partial Plagiarized

98% Unique Content

Total Words: 809 Total Characters: 5899 Plagiarized Sentences: 0.86 Unique Sentences: 42.14 (98%)

Content Checked for Plagiarism

Chapter 3: Software Requirement Specification
3.1 Introduction
3.1.1 Purpose
This document aims to define the objectives and scope of an AI-based Presentation Tool that utilizes gesture recognition and voice commands for hands-free presentation control. The tool will feature real-time gesture detection, voice integration, customizable slide controls, and offline operation. Designed to enhance the user experience, the system will make presentations more interactive, accessible, and intuitive, particularly for educators, professionals, and speakers. It also emphasizes user privacy and ease of use while integrating AI-powered features, such as speech-to-text and real-time translation, to ensure inclusivity and engagement for diverse audiences.

Why the Project Was Chosen
The project was selected to address the common challenges presenters face, such as the need to manually operate devices or stay close to a computer for slide transitions. By incorporating gesture recognition and voice commands, the tool allows presenters to move freely and engage naturally with their content. This approach enhances the overall presentation experience, making it more engaging and aligned with current trends in AI and human-computer interaction. The tool aims to be innovative and practical, offering solutions for classrooms, conferences, and professional settings.

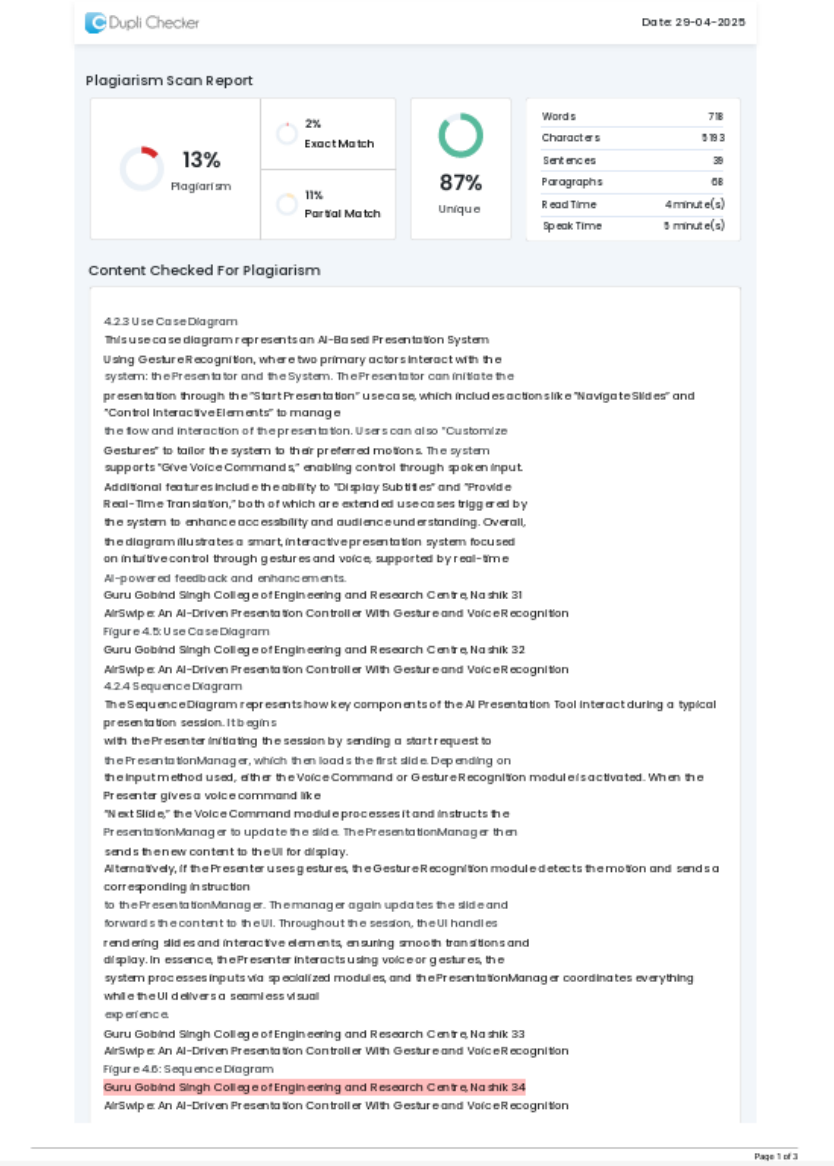
3.1.2 Intended Audience and Reading Suggestions
Educators and Teachers: Can utilize this tool in classrooms and seminars for dynamic, interactive presentations without needing to stay near a computer or use a remote control.

Corporate Professionals: Ideal for meetings, pitches, and conferences, offering seamless, hands-free control that enhances the professionalism and flow of presentations.

Public Speakers and Trainers: Provides freedom of movement and natural interaction with the audience, improving engagement and communication.

3.1.3 Project Scope
This project covers the design and development of an AI-driven presentation tool that enables hands-free slide control using gesture recognition and voice commands. The system will detect predefined gestures in real-time via a webcam, executing corresponding actions such as moving to the next or previous slide, pausing, or interacting with content. The system will also support voice commands for added flexibility and work offline. It will be compatible with popular presentation software. Key

Figure 10.3: Plagiarism Report for Chapter 3



Dupli Checker Date: 29-04-2025

Plagiarism Scan Report

13% Plagiarism

2% Exact Match

11% Partial Match

87% Unique

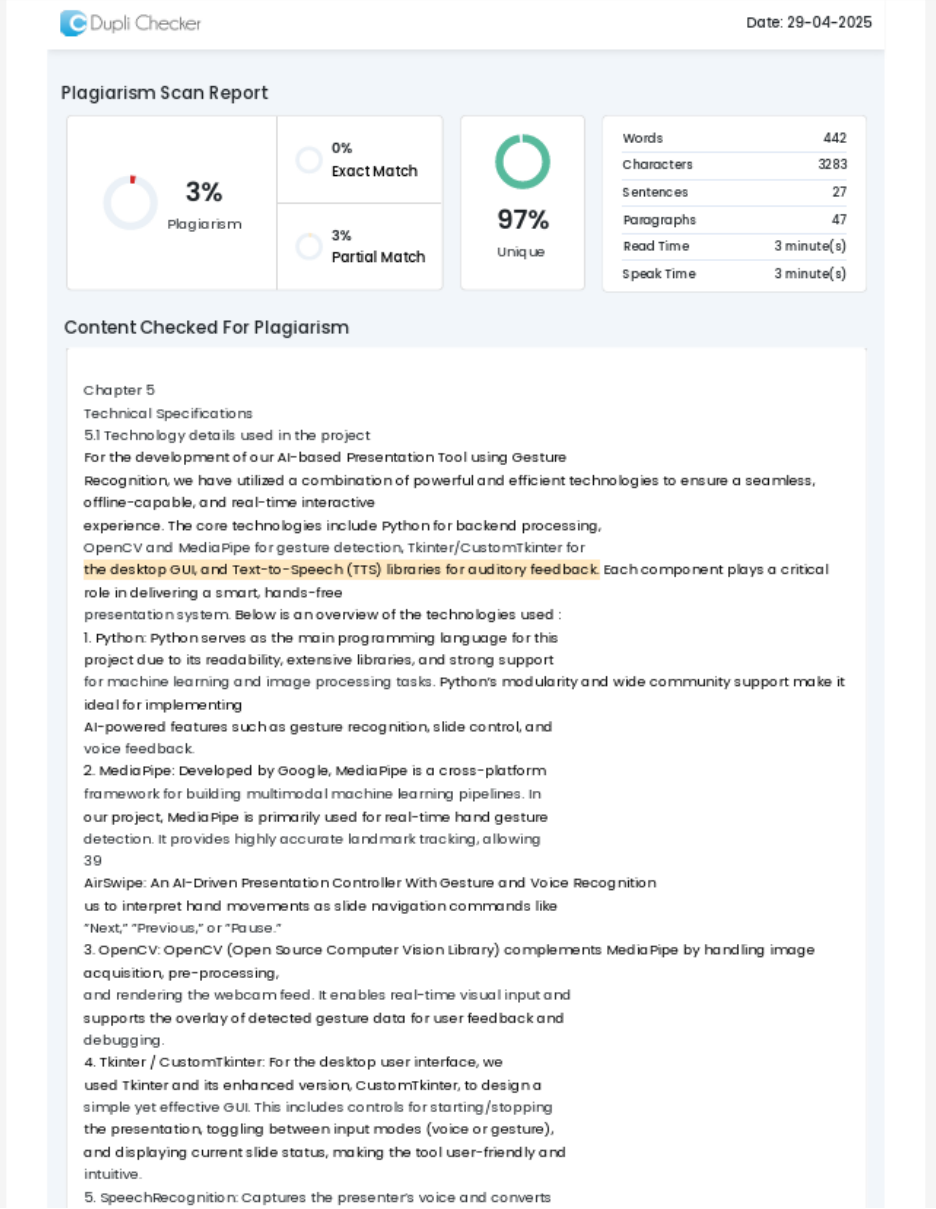
Words	718
Characters	5193
Sentences	39
Paragraphs	68
Read Time	4 minute(s)
Speak Time	5 minute(s)

Content Checked For Plagiarism

4.2.3 Use Case Diagram
This use case diagram represents an AI-Based Presentation System
Using Gesture Recognition, where two primary actors interact with the
system: the Presentator and the System. The Presentator can initiate the
presentation through the "Start Presentation" use case, which includes actions like "Navigate Slides" and
"Control Interactive Elements" to manage
the flow and interaction of the presentation. Users can also "Customize
Gestures" to tailor the system to their preferred motions. The system
supports "Give Voice Commands," enabling control through spoken input.
Additional features include the ability to "Display Subtitles" and "Provide
Real-Time Translation," both of which are extended use cases triggered by
the system to enhance accessibility and audience understanding. Overall,
the diagram illustrates a smart, interactive presentation system focused
on intuitive control through gestures and voice, supported by real-time
AI-powered feedback and enhancements.
Guru Gobind Singh College of Engineering and Research Centre, Nashik 31
AirSwipe: An AI-Driven Presentation Controller With Gesture and Voice Recognition
Figure 4.5: Use Case Diagram
Guru Gobind Singh College of Engineering and Research Centre, Nashik 32
AirSwipe: An AI-Driven Presentation Controller With Gesture and Voice Recognition
4.2.4 Sequence Diagram
The Sequence Diagram represents how key components of the AI Presentation Tool interact during a typical
presentation session. It begins
with the Presenter initiating the session by sending a start request to
the PresentationManager, which then loads the first slide. Depending on
the input method used, either the Voice Command or Gesture Recognition module is activated. When the
Presenter gives a voice command like
"Next Slide," the Voice Command module processes it and instructs the
PresentationManager to update the slide. The PresentationManager then
sends the new content to the UI for display.
Alternatively, if the Presenter uses gestures, the Gesture Recognition module detects the motion and sends a
corresponding instruction
to the PresentationManager. The manager again updates the slide and
forwards the content to the UI. Throughout the session, the UI handles
rendering slides and interactive elements, ensuring smooth transitions and
display. In essence, the Presenter interacts using voice or gestures, the
system processes inputs via specialized modules, and the PresentationManager coordinates everything
while the UI delivers a seamless visual
experience.
Guru Gobind Singh College of Engineering and Research Centre, Nashik 33
AirSwipe: An AI-Driven Presentation Controller With Gesture and Voice Recognition
Figure 4.6: Sequence Diagram
Guru Gobind Singh College of Engineering and Research Centre, Nashik 34
AirSwipe: An AI-Driven Presentation Controller With Gesture and Voice Recognition

Page 1 of 3

Figure 10.4: Plagiarism Report for Chapter 4



Dupli Checker

Date: 29-04-2025

Plagiarism Scan Report

3% Plagiarism

0% Exact Match

3% Partial Match

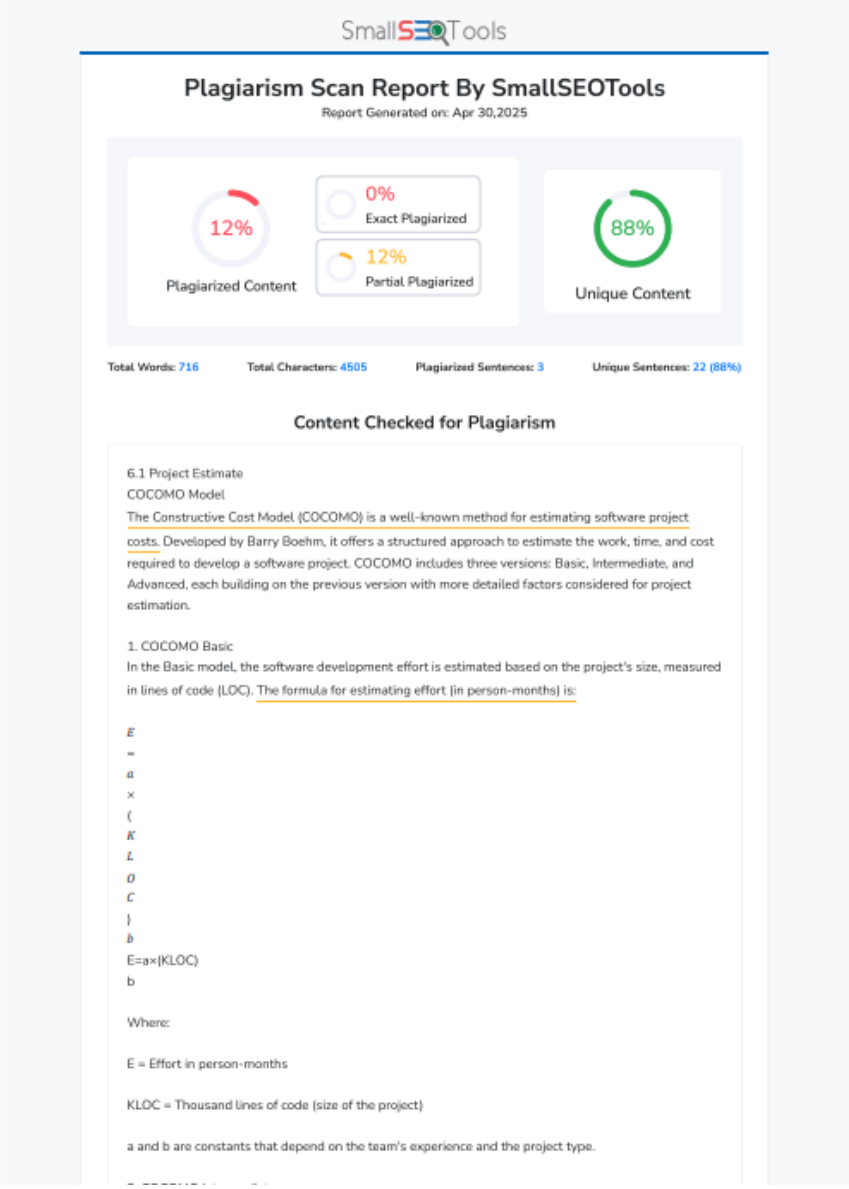
97% Unique

Words	442
Characters	3283
Sentences	27
Paragraphs	47
Read Time	3 minute(s)
Speak Time	3 minute(s)

Content Checked For Plagiarism

Chapter 5
Technical Specifications
5.1 Technology details used in the project
For the development of our AI-based Presentation Tool using Gesture
Recognition, we have utilized a combination of powerful and efficient technologies to ensure a seamless, offline-capable, and real-time interactive
experience. The core technologies include Python for backend processing,
OpenCV and MediaPipe for gesture detection, Tkinter/CustomTkinter for
the desktop GUI, and Text-to-Speech (TTS) libraries for auditory feedback. Each component plays a critical role in delivering a smart, hands-free
presentation system. Below is an overview of the technologies used :
1. Python: Python serves as the main programming language for this
project due to its readability, extensive libraries, and strong support
for machine learning and image processing tasks. Python's modularity and wide community support make it ideal for implementing
AI-powered features such as gesture recognition, slide control, and
voice feedback.
2. MediaPipe: Developed by Google, MediaPipe is a cross-platform
framework for building multimodal machine learning pipelines. In
our project, MediaPipe is primarily used for real-time hand gesture
detection. It provides highly accurate landmark tracking, allowing
39
AirSwipe: An AI-Driven Presentation Controller With Gesture and Voice Recognition
us to interpret hand movements as slide navigation commands like
"Next," "Previous," or "Pause."
3. OpenCV: OpenCV (Open Source Computer Vision Library) complements MediaPipe by handling image acquisition, pre-processing,
and rendering the webcam feed. It enables real-time visual input and
supports the overlay of detected gesture data for user feedback and
debugging.
4. Tkinter / CustomTkinter: For the desktop user interface, we
used Tkinter and its enhanced version, CustomTkinter, to design a
simple yet effective GUI. This includes controls for starting/stopping
the presentation, toggling between input modes (voice or gesture),
and displaying current slide status, making the tool user-friendly and
intuitive.
5. SpeechRecognition: Captures the presenter's voice and converts

Figure 10.5: Plagiarism Report for Chapter 5



SmallSEOTools

Plagiarism Scan Report By SmallSEOTools

Report Generated on: Apr 30,2025

12% Plagiarized Content

0% Exact Plagiarized

12% Partial Plagiarized

88% Unique Content

Total Words: 716 Total Characters: 4505 Plagiarized Sentences: 3 Unique Sentences: 22 (88%)

Content Checked for Plagiarism

6.1 Project Estimate
COCOMO Model
The Constructive Cost Model (COCOMO) is a well-known method for estimating software project costs. Developed by Barry Boehm, it offers a structured approach to estimate the work, time, and cost required to develop a software project. COCOMO includes three versions: Basic, Intermediate, and Advanced, each building on the previous version with more detailed factors considered for project estimation.

1. COCOMO Basic
In the Basic model, the software development effort is estimated based on the project's size, measured in lines of code (LOC). The formula for estimating effort (in person-months) is:

$$E = a \times (KLOC)^b$$

E=a×(KLOC)
b

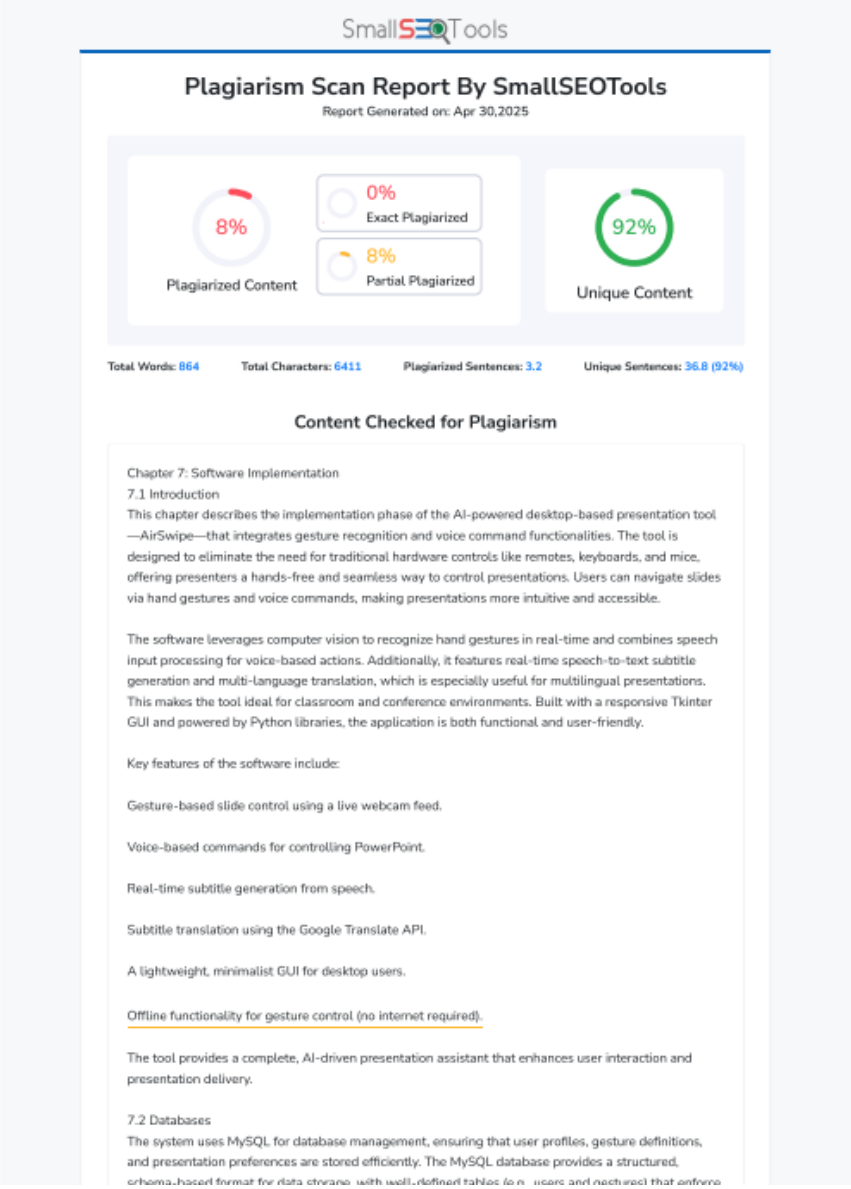
Where:

E = Effort in person-months

KLOC = Thousand lines of code (size of the project)

a and b are constants that depend on the team's experience and the project type.

Figure 10.6: Plagiarism Report for Chapter 6



SmallSEOTools

Plagiarism Scan Report By SmallSEOTools

Report Generated on: Apr 30,2025

8% Plagiarized Content

0% Exact Plagiarized

8% Partial Plagiarized

92% Unique Content

Total Words: 864 Total Characters: 6411 Plagiarized Sentences: 3.2 Unique Sentences: 36.8 (92%)

Content Checked for Plagiarism

Chapter 7: Software Implementation

7.1 Introduction

This chapter describes the implementation phase of the AI-powered desktop-based presentation tool —AirSwipe—that integrates gesture recognition and voice command functionalities. The tool is designed to eliminate the need for traditional hardware controls like remotes, keyboards, and mice, offering presenters a hands-free and seamless way to control presentations. Users can navigate slides via hand gestures and voice commands, making presentations more intuitive and accessible.

The software leverages computer vision to recognize hand gestures in real-time and combines speech input processing for voice-based actions. Additionally, it features real-time speech-to-text subtitle generation and multi-language translation, which is especially useful for multilingual presentations. This makes the tool ideal for classroom and conference environments. Built with a responsive Tkinter GUI and powered by Python libraries, the application is both functional and user-friendly.

Key features of the software include:

Gesture-based slide control using a live webcam feed.

Voice-based commands for controlling PowerPoint.

Real-time subtitle generation from speech.

Subtitle translation using the Google Translate API.

A lightweight, minimalist GUI for desktop users.

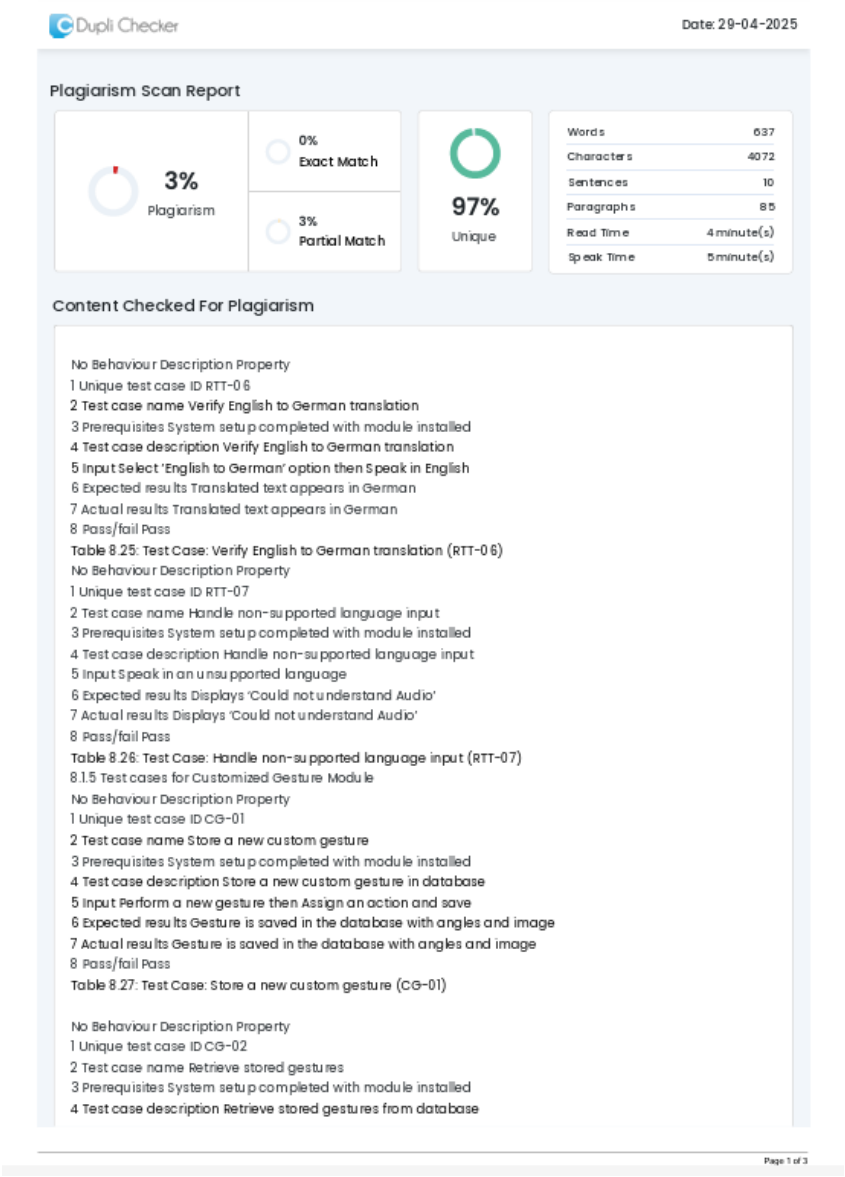
Offline functionality for gesture control (no internet required).

The tool provides a complete, AI-driven presentation assistant that enhances user interaction and presentation delivery.

7.2 Databases

The system uses MySQL for database management, ensuring that user profiles, gesture definitions, and presentation preferences are stored efficiently. The MySQL database provides a structured, schema-based format for data storage, with well-defined tables (e.g., users and gestures) that enforce

Figure 10.7: Plagiarism Report for Chapter 7



Dupli Checker Date: 29-04-2025

Plagiarism Scan Report

3% Plagiarism

0% Exact Match

3% Partial Match

97% Unique

Words	637
Characters	4072
Sentences	10
Paragraphs	85
Read Time	4 minute(s)
Speak Time	5 minute(s)

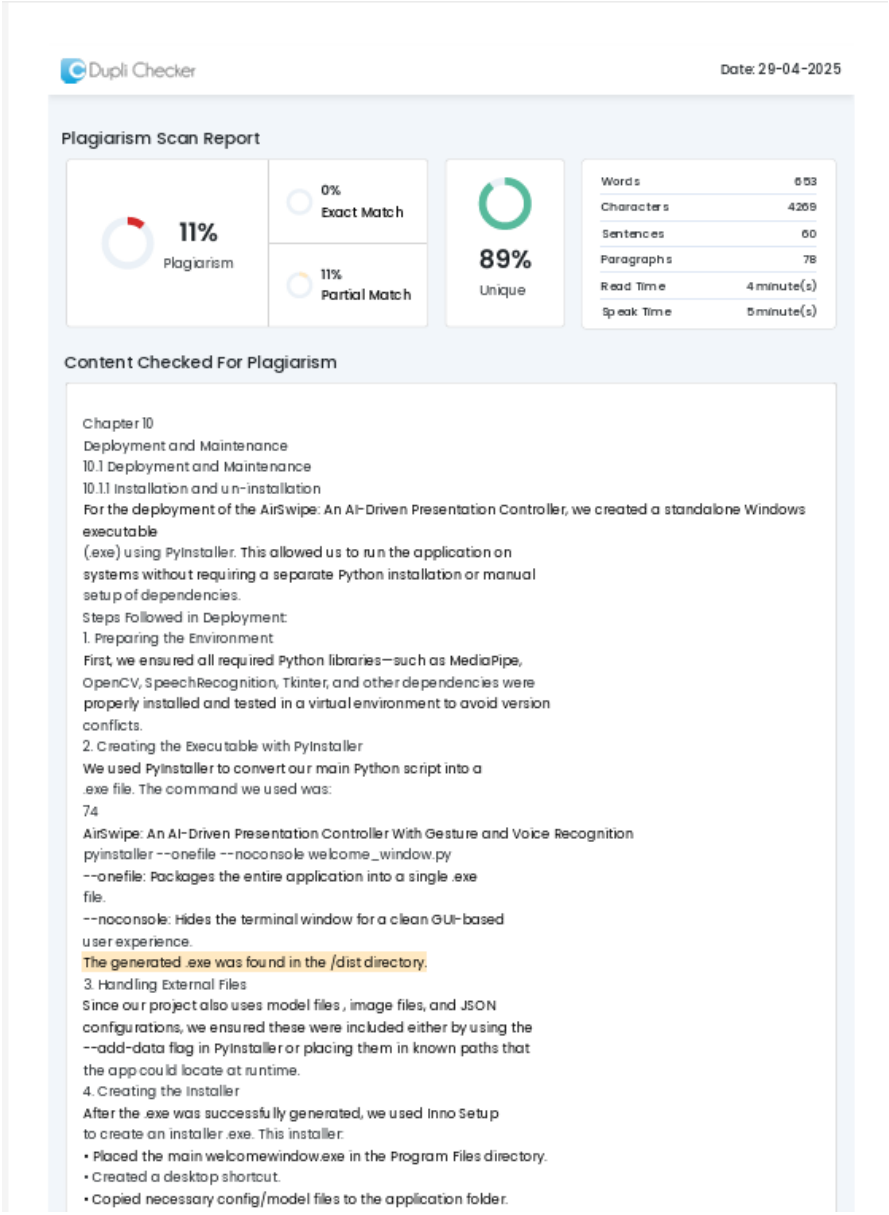
Content Checked For Plagiarism

No Behaviour Description Property
1 Unique test case ID RTT-06
2 Test case name Verify English to German translation
3 Prerequisites System setup completed with module installed
4 Test case description Verify English to German translation
5 Input Select 'English to German' option then Speak in English
6 Expected results Translated text appears in German
7 Actual results Translated text appears in German
8 Pass/fail Pass
Table 8.25: Test Case: Verify English to German translation (RTT-06)
No Behaviour Description Property
1 Unique test case ID RTT-07
2 Test case name Handle non-supported language input
3 Prerequisites System setup completed with module installed
4 Test case description Handle non-supported language input
5 Input Speak in an unsupported language
6 Expected results Displays 'Could not understand Audio'
7 Actual results Displays 'Could not understand Audio'
8 Pass/fail Pass
Table 8.26: Test Case: Handle non-supported language input (RTT-07)
8.1.5 Test cases for Customized Gesture Module
No Behaviour Description Property
1 Unique test case ID CG-01
2 Test case name Store a new custom gesture
3 Prerequisites System setup completed with module installed
4 Test case description Store a new custom gesture in database
5 Input Perform a new gesture then Assign an action and save
6 Expected results Gesture is saved in the database with angles and image
7 Actual results Gesture is saved in the database with angles and image
8 Pass/fail Pass
Table 8.27: Test Case: Store a new custom gesture (CG-01)

No Behaviour Description Property
1 Unique test case ID CG-02
2 Test case name Retrieve stored gestures
3 Prerequisites System setup completed with module installed
4 Test case description Retrieve stored gestures from database

Page 1 of 3

Figure 10.8: Plagiarism Report for Chapter 8



Dupli Checker

Date: 29-04-2025

Plagiarism Scan Report

11% Plagiarism

0% Exact Match

11% Partial Match

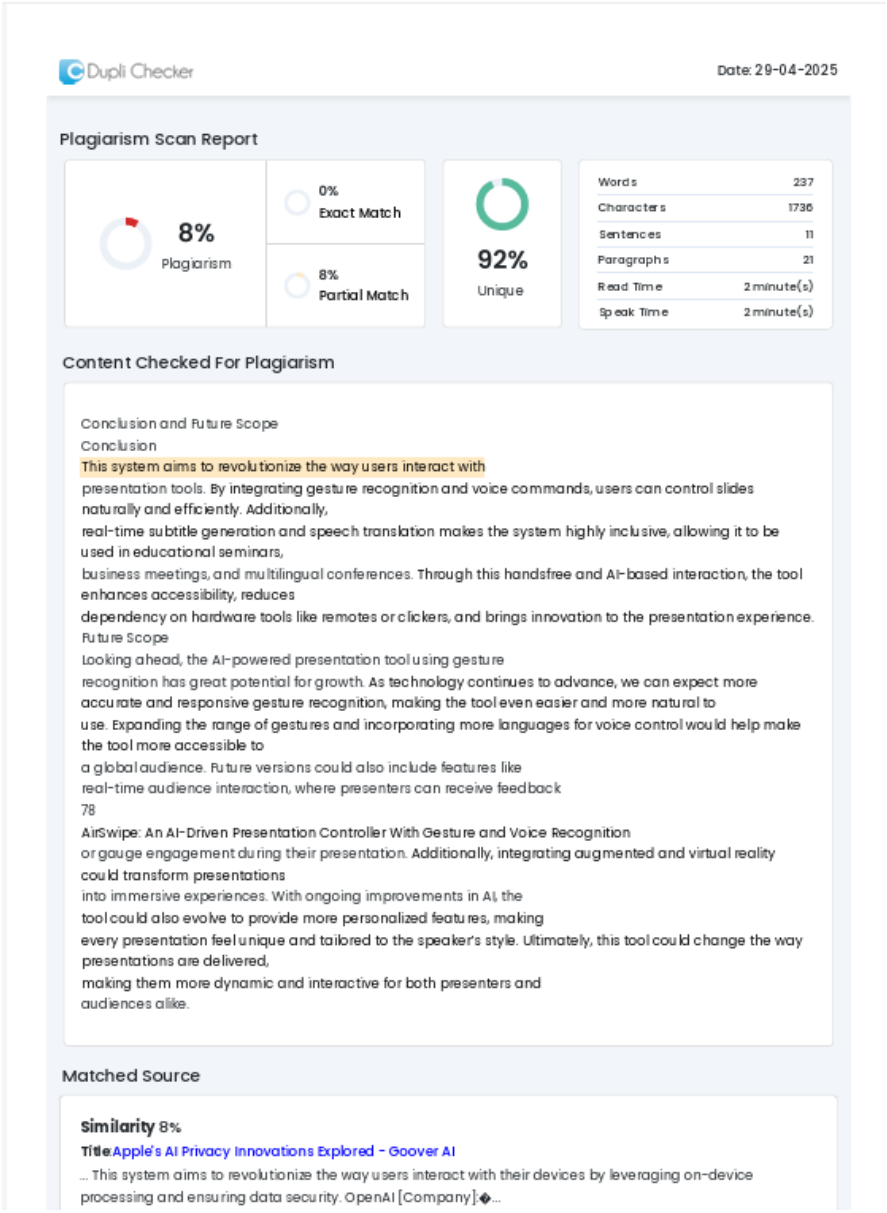
89% Unique

Words	653
Characters	4269
Sentences	60
Paragraphs	78
Read Time	4 minute(s)
Speak Time	5 minute(s)

Content Checked For Plagiarism

Chapter 10
Deployment and Maintenance
10.1 Deployment and Maintenance
10.1.1 Installation and un-installation
For the deployment of the AirSwipe: An AI-Driven Presentation Controller, we created a standalone Windows executable
(.exe) using PyInstaller. This allowed us to run the application on
systems without requiring a separate Python installation or manual
setup of dependencies.
Steps Followed in Deployment:
1. Preparing the Environment
First, we ensured all required Python libraries—such as MediaPipe,
OpenCV, SpeechRecognition, Tkinter, and other dependencies were
properly installed and tested in a virtual environment to avoid version
conflicts.
2. Creating the Executable with PyInstaller
We used PyInstaller to convert our main Python script into a
.exe file. The command we used was:
74
AirSwipe: An AI-Driven Presentation Controller With Gesture and Voice Recognition
pyinstaller --onefile --noconsole welcome_window.py
--onefile: Packages the entire application into a single .exe
file.
--noconsole: Hides the terminal window for a clean GUI-based
user experience.
The generated .exe was found in the /dist directory.
3. Handling External Files
Since our project also uses model files , image files, and JSON
configurations, we ensured these were included either by using the
--add-data flag in PyInstaller or placing them in known paths that
the app could locate at runtime.
4. Creating the Installer
After the .exe was successfully generated, we used Inno Setup
to create an installer .exe. This installer:
• Placed the main welcomewindow.exe in the Program Files directory.
• Created a desktop shortcut.
• Copied necessary config/model files to the application folder.

Figure 10.9: Plagiarism Report for Chapter 9



Dupli Checker

Date: 29-04-2025

Plagiarism Scan Report

8% Plagiarism

0% Exact Match

8% Partial Match

92% Unique

Words	237
Characters	1736
Sentences	11
Paragraphs	21
Read Time	2 minute(s)
Speak Time	2 minute(s)

Content Checked For Plagiarism

Conclusion and Future Scope
Conclusion
This system aims to revolutionize the way users interact with
presentation tools. By integrating gesture recognition and voice commands, users can control slides
naturally and efficiently. Additionally,
real-time subtitle generation and speech translation makes the system highly inclusive, allowing it to be
used in educational seminars,
business meetings, and multilingual conferences. Through this handsfree and AI-based interaction, the tool
enhances accessibility, reduces
dependency on hardware tools like remotes or clickers, and brings innovation to the presentation experience.
Future Scope
Looking ahead, the AI-powered presentation tool using gesture
recognition has great potential for growth. As technology continues to advance, we can expect more
accurate and responsive gesture recognition, making the tool even easier and more natural to
use. Expanding the range of gestures and incorporating more languages for voice control would help make
the tool more accessible to
a global audience. Future versions could also include features like
real-time audience interaction, where presenters can receive feedback
78
AirSwipe: An AI-Driven Presentation Controller With Gesture and Voice Recognition
or gauge engagement during their presentation. Additionally, integrating augmented and virtual reality
could transform presentations
into immersive experiences. With ongoing improvements in AI, the
tool could also evolve to provide more personalized features, making
every presentation feel unique and tailored to the speaker's style. Ultimately, this tool could change the way
presentations are delivered,
making them more dynamic and interactive for both presenters and
audiences alike.

Matched Source

Similarity 8%

Title: Apple's AI Privacy Innovations Explored - Goover AI

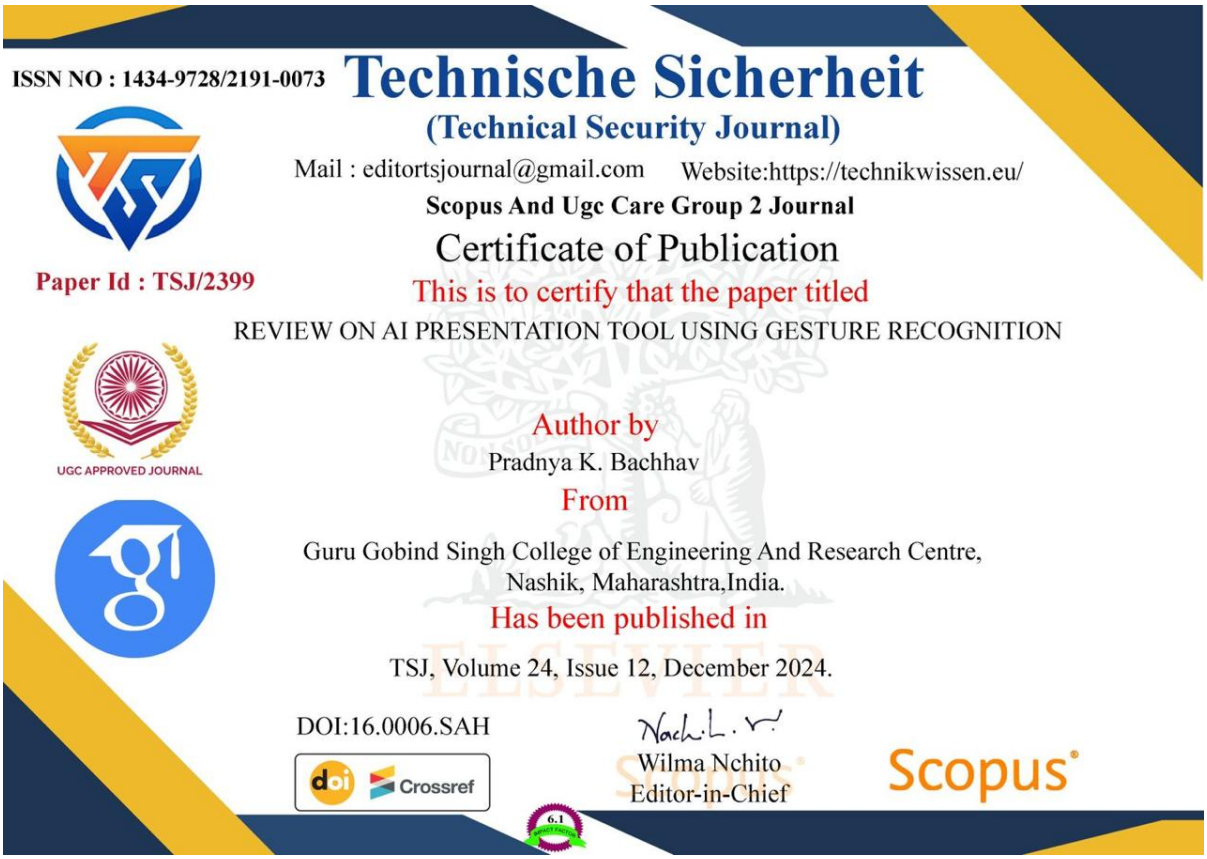
... This system aims to revolutionize the way users interact with their devices by leveraging on-device processing and ensuring data security. OpenAI [Company]:...

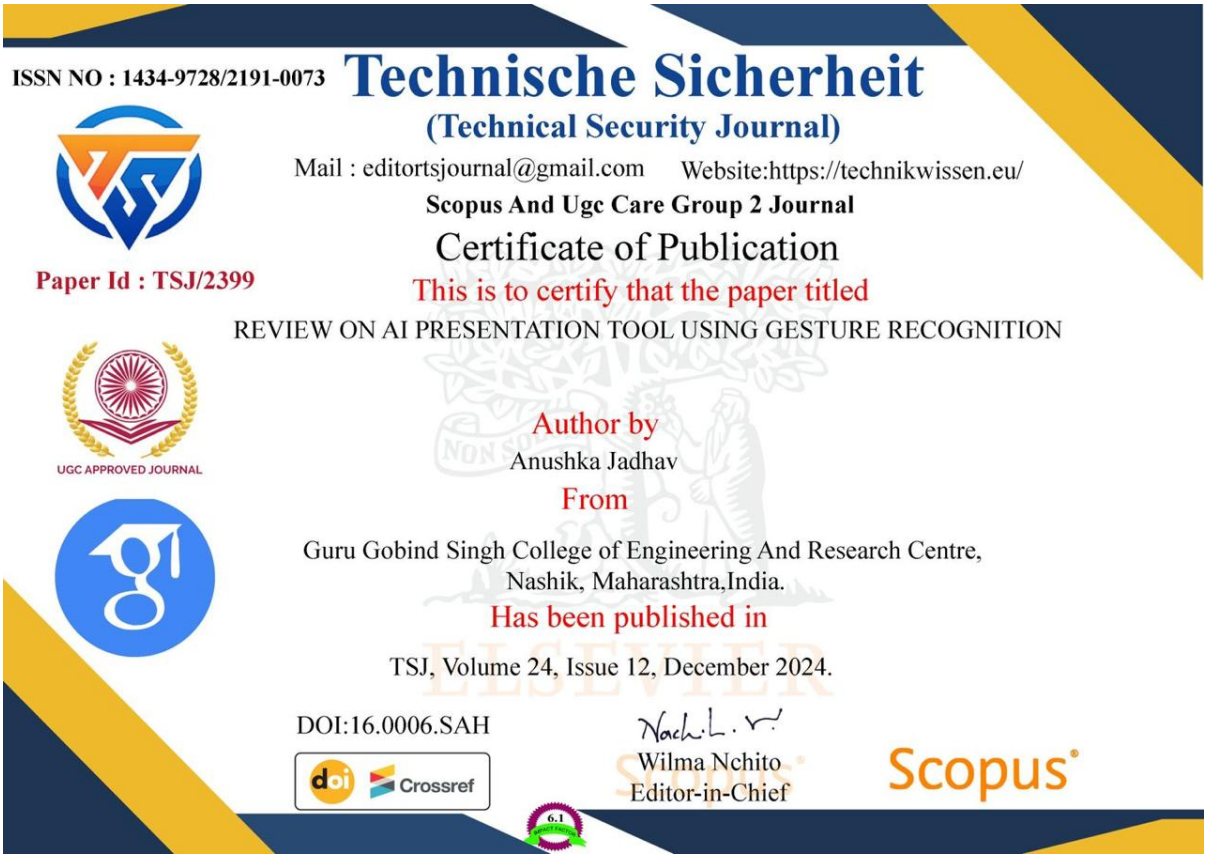
Figure 10.10: Plagiarism Report for Chapter 10

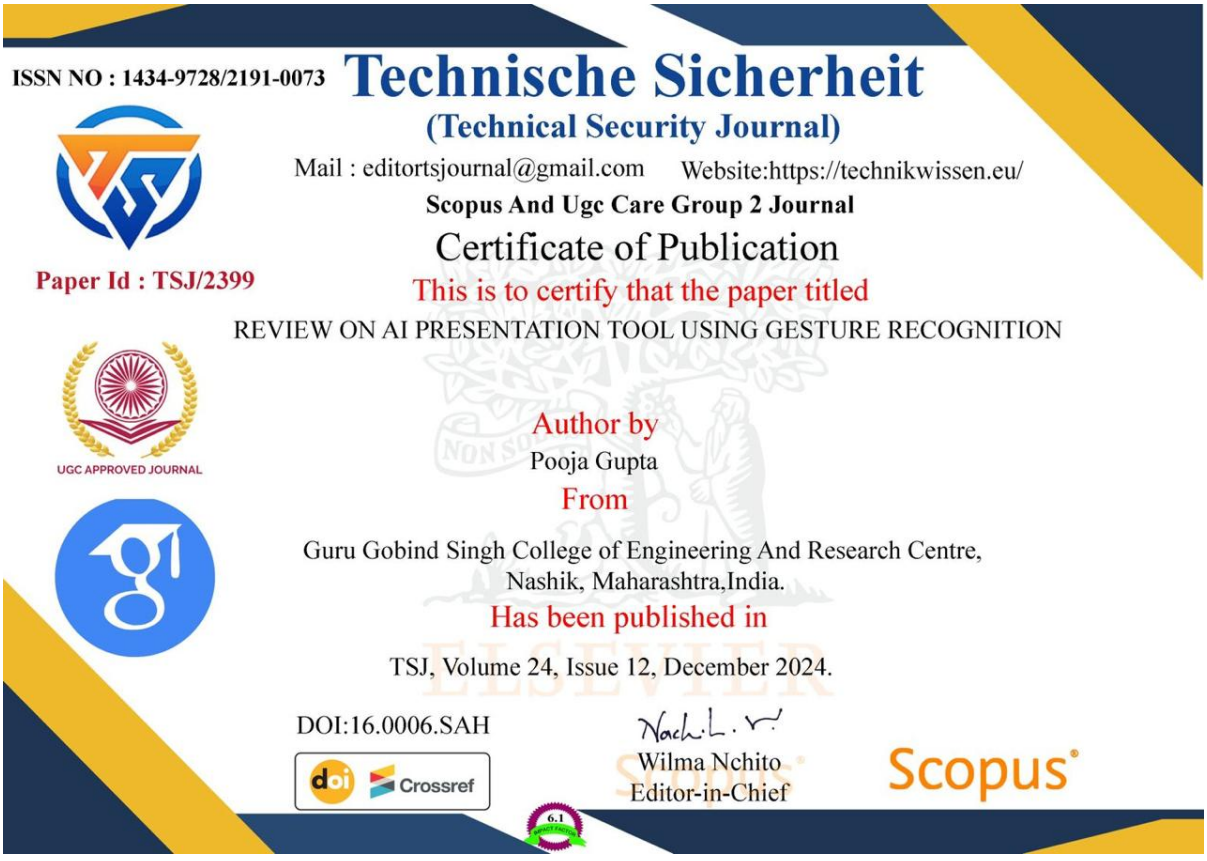
Paper Publication and Certificate Details

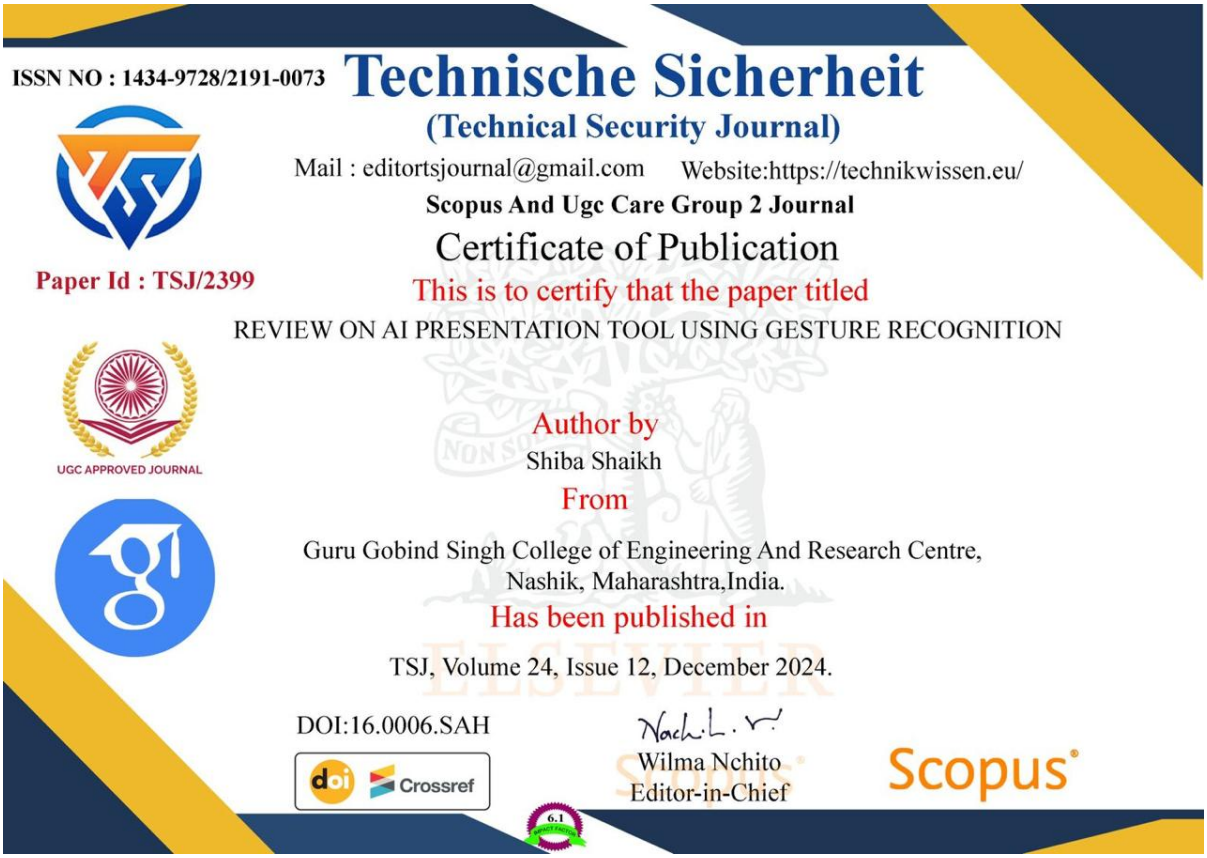
Published Paper

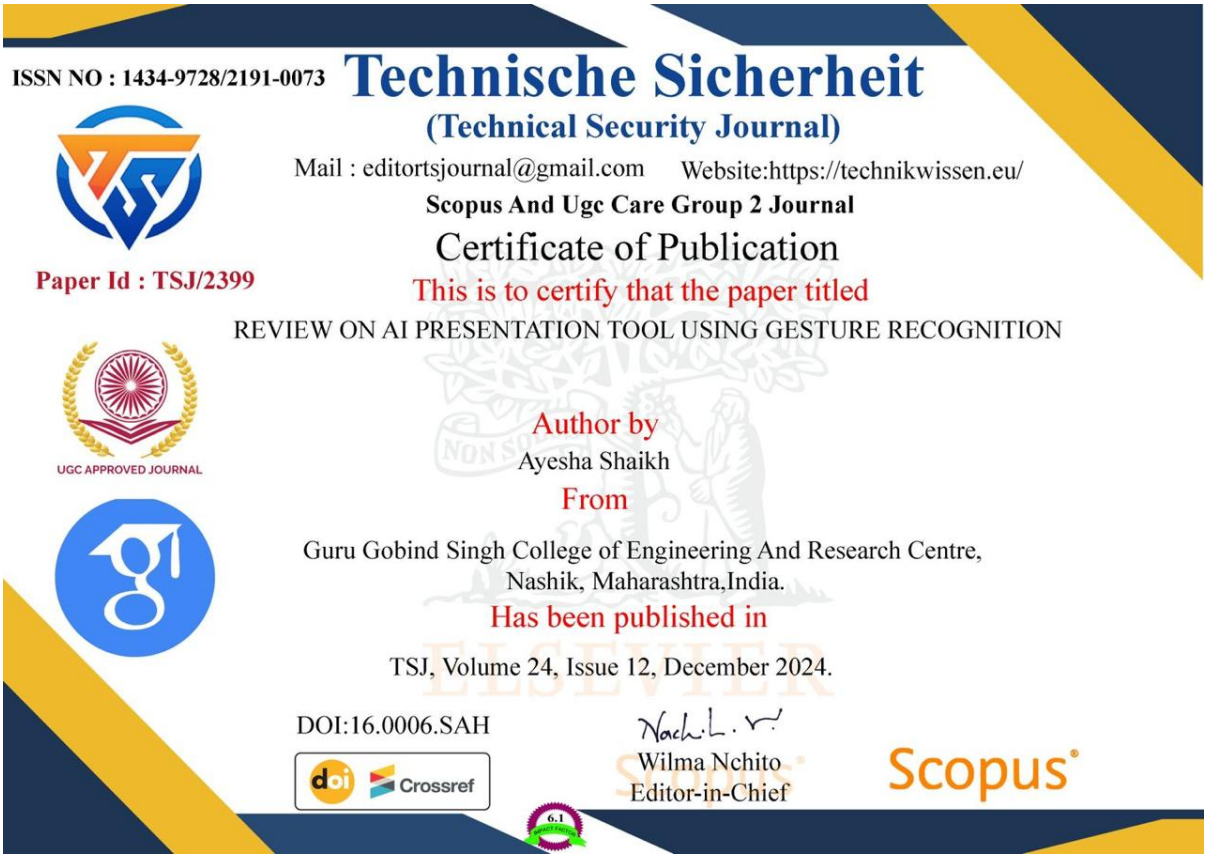
Published paper in Technical Security Journal on "Review on AI Presentation Tool Using Gesture Recognition"
Paper ID: TSJ/2399, ISSN-1434-9728/2191-0073.




ISSN NO : 1434-9728/2191-0073
Technische Sicherheit
(Technical Security Journal)
Mail : editortsjournal@gmail.com
Website:https://technikwissen.eu/
Scopus And Ugc Care Group 2 Journal
Paper Id : TSJ/2399
Certificate of Publication
This is to certify that the paper titled
REVIEW ON AI PRESENTATION TOOL USING GESTURE RECOGNITION
UGC APPROVED JOURNAL
Author by
Anushka Jadhav
From
Guru Gobind Singh College of Engineering And Research Centre,
Nashik, Maharashtra,India.
Has been published in
TSJ, Volume 24, Issue 12, December 2024.
DOI:16.0006.SAH
Crossref
Wilma Nchito
Editor-in-Chief
Scopus


ISSN NO : 1434-9728/2191-0073
Technische Sicherheit
(Technical Security Journal)
Mail : editortsjournal@gmail.com
Website:https://technikwissen.eu/
Scopus And Ugc Care Group 2 Journal
Paper Id : TSJ/2399
Certificate of Publication
This is to certify that the paper titled
REVIEW ON AI PRESENTATION TOOL USING GESTURE RECOGNITION
UGC APPROVED JOURNAL
Author by
Pooja Gupta
From
Guru Gobind Singh College of Engineering And Research Centre,
Nashik, Maharashtra,India.
Has been published in
TSJ, Volume 24, Issue 12, December 2024.
DOI:16.0006.SAH
Crossref
Wilma Nchito
Editor-in-Chief
Scopus


ISSN NO : 1434-9728/2191-0073
Technische Sicherheit
(Technical Security Journal)
Mail : editortsjournal@gmail.com
Website:https://technikwissen.eu/
Scopus And Ugc Care Group 2 Journal
Paper Id : TSJ/2399
Certificate of Publication
This is to certify that the paper titled
REVIEW ON AI PRESENTATION TOOL USING GESTURE RECOGNITION
UGC APPROVED JOURNAL
Author by
Shiba Shaikh
From
Guru Gobind Singh College of Engineering And Research Centre,
Nashik, Maharashtra,India.
Has been published in
TSJ, Volume 24, Issue 12, December 2024.
DOI:16.0006.SAH
Crossref
Wilma Nchito
Editor-in-Chief
Scopus


ISSN NO : 1434-9728/2191-0073
Technische Sicherheit
(Technical Security Journal)
Mail : editortsjournal@gmail.com
Website:https://technikwissen.eu/
Scopus And Ugc Care Group 2 Journal
Paper Id : TSJ/2399
Certificate of Publication
This is to certify that the paper titled
REVIEW ON AI PRESENTATION TOOL USING GESTURE RECOGNITION
UGC APPROVED JOURNAL
Author by
Ayesha Shaikh
From
Guru Gobind Singh College of Engineering And Research Centre,
Nashik, Maharashtra,India.
Has been published in
TSJ, Volume 24, Issue 12, December 2024.
DOI:16.0006.SAH
Crossref
Wilma Nchito
Editor-in-Chief
Scopus

Published paper in International Research Journal on Advanced Engineering Hub(IRJAEH) on "AI Presentation Tool Using Gesture Recognition", ISSN-2584-2137.


e-ISSN No: 2584-2137
OPEN ACCESS
INTERNATIONAL RESEARCH JOURNAL ON ADVANCED ENGINEERING HUB (IRJAEH)
Email: editor@irjaeh.com
Available online at : https://irjaeh.com/index.php/journal
CERTIFICATE OF PUBLICATION
IRJAEH Is Hereby Awarding This Certificate To
Ayesha Shaikh
In Recognition of The Publication of The Manuscript Entitled
AI Presentation Tool Using Gesture Recognition
Published In Volume 03 Issue 03 March 2025.
Dr. M. Subramanian
Editor in Chief, IRJAEH,
Coimbatore,India


e-ISSN No: 2584-2137
OPEN ACCESS
INTERNATIONAL RESEARCH JOURNAL ON ADVANCED ENGINEERING HUB (IRJAEH)
Email: editor@irjaeh.com
Available online at : https://irjaeh.com/index.php/journal
CERTIFICATE OF PUBLICATION
IRJAEH Is Hereby Awarding This Certificate To
Pradnya K. Bachhav
In Recognition of The Publication of The Manuscript Entitled
AI Presentation Tool Using Gesture Recognition
Published In Volume 03 Issue 03 March 2025.
Dr. M. Subramanian
Editor in Chief, IRJAEH,
Coimbatore,India


e-ISSN No: 2584-2137
OPEN ACCESS
INTERNATIONAL RESEARCH JOURNAL ON ADVANCED ENGINEERING HUB (IRJAEH)
Email: editor@irjaeh.com
Available online at : https://irjaeh.com/index.php/journal
CERTIFICATE OF PUBLICATION
IRJAEH Is Hereby Awarding This Certificate To
Anushka Jadhav
In Recognition of The Publication of The Manuscript Entitled
AI Presentation Tool Using Gesture Recognition
Published In Volume 03 Issue 03 March 2025.
Dr. M. Subramanian
Editor in Chief, IRJAEH,
Coimbatore,India


e-ISSN No: 2584-2137
OPEN ACCESS
INTERNATIONAL RESEARCH JOURNAL ON ADVANCED ENGINEERING HUB (IRJAEH)
Email: editor@irjaeh.com
Available online at : https://irjaeh.com/index.php/journal
CERTIFICATE OF PUBLICATION
IRJAEH Is Hereby Awarding This Certificate To
Shiba Shaikh
In Recognition of The Publication of The Manuscript Entitled
AI Presentation Tool Using Gesture Recognition
Published In Volume 03 Issue 03 March 2025.
Dr. M. Subramanian
Editor in Chief, IRJAEH,
Coimbatore,India

Conference Paper

International Conference on Engineering Scienece and Management(ISCEM) 2025

CERTIFICATE

OF PARTICIPATION

This is to Certify that

Ms. Pooja Gupta

Ug Student,Department of Computer Engineering, Guru Gobind Singh College of Engineering and Research Centre, Nashik.

For attending and giving an oral presentation on the paper entitled

AI Presentation Tool Using Gesture Recognition

at the "International Conference on Engineering, Science and Management (ICESM) 2025" - Organized by Department of Computer Engineering & Electrical Engineering, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India & Event Organizer: RSP Conference Hub, Coimbatore, Tamil Nadu, India on 11th & 12th March 2025.

Dr. Rahul Agrawal
HoD, Department of Electrical Engineering, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India

Prof. S. G. Shukla
HoD, Department of Computer Engineering, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India

Dr. N.G. Nikam
Principal, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India

GGSF & RSP ICESM 2025-538

CERTIFICATE

OF PARTICIPATION

This is to Certify that

Ms. Ayesha Shaikh

Ug Student,Department of Computer Engineering, Guru Gobind Singh College of Engineering and Research Centre, Nashik.

For attending and giving an oral presentation on the paper entitled

AI Presentation Tool Using Gesture Recognition

at the "International Conference on Engineering, Science and Management (ICESM) 2025" - Organized by Department of Computer Engineering & Electrical Engineering, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India & Event Organizer: RSP Conference Hub, Coimbatore, Tamil Nadu, India on 11th & 12th March 2025.

Dr. Rahul Agrawal
HoD, Department of Electrical Engineering, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India

Prof. S. G. Shukla
HoD, Department of Computer Engineering, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India

Dr. N.G. Nikam
Principal, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India

GGSF & RSP ICESM 2025-537

CERTIFICATE

OF PARTICIPATION

This is to Certify that

Prof. Pradnya Bachhav

Assistant Professor,Department of Computer Engineering, Guru Gobind Singh College of Engineering and Research Centre, Nashik.

For attending and giving an oral presentation on the paper entitled

AI Presentation Tool Using Gesture Recognition

at the "International Conference on Engineering, Science and Management (ICESM) 2025" - Organized by Department of Computer Engineering & Electrical Engineering, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India & Event Organizer: RSP Conference Hub, Coimbatore, Tamil Nadu, India on 11th & 12th March 2025.

Dr. Rahul Agrawal
HoD, Department of Electrical Engineering, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India

Prof. S. G. Shukla
HoD, Department of Computer Engineering, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India

Dr. N.G. Nikam
Principal, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India

GGSF & RSP ICESM 2025-193

CERTIFICATE

OF PARTICIPATION

This is to Certify that

Ms. Anushka Jadhav(Ug Student)

Department of Computer Engineering, Guru Gobind Singh College of Engineering and Research Centre, Nashik.

For attending and giving an oral presentation on the paper entitled

AI Presentation Tool Using Gesture Recognition

at the "International Conference on Engineering, Science and Management (ICESM) 2025" - Organized by Department of Computer Engineering & Electrical Engineering, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India & Event Organizer: RSP Conference Hub, Coimbatore, Tamil Nadu, India on 11th & 12th March 2025.

Dr. Rahul Agrawal
HoD, Department of Electrical Engineering, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India

Prof. S. G. Shukla
HoD, Department of Computer Engineering, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India

Dr. N.G. Nikam
Principal, Guru Gobind Singh College of Engineering & Research Centre, Nashik, Maharashtra, India

GGSF & RSP ICESM 2025-207

Project Competitions

1. Startup Arena(National Entrepreneurship Challenge 2024)


NATIONAL ENTREPRENEURSHIP CHALLENGE 2024
NEC
CERTIFICATE
OF ACHIEVEMENT
THIS CERTIFICATE IS REWARDED TO
Ms. Pooja · G. Gupta
of GCOERC in recognition of his/her participation in StartUp Arena event organized on October 1, 2024, showcasing entrepreneurial skills and innovation.
PROF.PRAMOD PATIL
Faculty advisor IEDCell
AICTE-F.NO. WESTERN/2013/ 1-1404766831
DR.NEELKANTH G.NIKAM
Principal GCOERC


NATIONAL ENTREPRENEURSHIP CHALLENGE 2024
NEC
CERTIFICATE
OF ACHIEVEMENT
THIS CERTIFICATE IS REWARDED TO
Ms. Shiba Anwar Shaikh
of GCOERC in recognition of his/her participation in StartUp Arena event organized on October 1, 2024, showcasing entrepreneurial skills and innovation.
PROF.PRAMOD PATIL
Faculty advisor IEDCell
AICTE-F.NO. WESTERN/2013/
DR.NEELKANTH G.NIKAM
Principal GCOERC

2. Anveshana 2024-25 Regional Level Science and Engineering Fair.

3. Techguru (2025)


Presents
TECHGURU 2K25
Media Partner
Associate Sponsors
Guru Gobind Singh Foundation's
Guru Gobind Singh College of Engineering and Research Centre, Nashik
TECHGURU 2K25
Certificate of Participation
This Certificate is proudly awarded to
Mr/ Mrs. Pooja Gupta
From Guru Gobind Singh Clg of Engineering
for Participation In Project competition under TechGuru 2K25
at Guru Gobind Singh College of Engineering and Research Centre,
Nashik on 28th-29th March 2025.
Congratulations
Dr. Sunil More
(Overall Coordinator)
Dr. C.D.Mohod
(Academic Dean)
Dr. Neelkanth G. Nikam
(Principal GCOERC)


Presents
TECHGURU 2K25
Media Partner
Associate Sponsors
Guru Gobind Singh Foundation's
Guru Gobind Singh College of Engineering and Research Centre, Nashik
TECHGURU 2K25
Certificate of Participation
This Certificate is proudly awarded to
Mr/ Mrs. Shiva shaikh
From Guru Gobind Singh Clg of Engineering
for Participation In Project competition under TechGuru 2K25
at Guru Gobind Singh College of Engineering and Research Centre,
Nashik on 28th-29th March 2025.
Congratulations
Dr. Sunil More
(Overall Coordinator)
Dr. C.D.Mohod
(Academic Dean)
Dr. Neelkanth G. Nikam
(Principal GCOERC)


Presents
TECHGURU 2K25
Media Partner
&
Associate Sponsors
Guru Gobind Singh Foundation's
Guru Gobind Singh College of Engineering and
Research Centre, Nashik
TECHGURU 2K25
Certificate of Participation
This Certificate is proudly awarded to
Mr/ Mrs. Ayesha shaikh
From Guru Gobind Singh Clg of Engineering
for Participation In Project Competition under TechGuru 2K25
at Guru Gobind Singh College of Engineering and Research Centre,
Nashik on 28th-29th March 2025.
Congratulations
Dr. Sunil More
(Overall Coordinator)
Dr. C.D.Mohod
(Academic Dean)
Dr. Neelkanth G. Nikam
(Principal GCOERC)