**GESTURE CONTROLLED PRESENTATION SYSTEM WITH SPEECH RECOGNITION AND WEB INTERACTION**

A Major Project report submitted

in partial fulfillment of requirement for the award of degree

**BACHELOR OF TECHNOLOGY**

in

**SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE**

by

**MOHD AMAAN UR RAHMAN (2103A51057)**

**THATIPELLI MANIKANTA (2103A51073)**

**DUGYALA VIVEK (2103A51088)**

**GADDAM VARSHITH (2103A51089)**

**KOTTE NAVEEN (2103A51267)**

**K RAGHAVENDRA CHARY (2103A51268)**

Under the guidance of

**Mr. Gotte Ranjith Kumar**

Assistant Professor,

School of CS&AI.



SR University, Ananthsagar,Warangal,Telagnana-506371

**SR University**

Ananthasagar, Warangal.

**

**CERTIFICATE**

This is to certify that this project entitled **“GESTURE CONTROLLED PRESENTATION SYSTEM WITH SPEECH RECOGNITION**" is the bonafide work carried out by **MOHD AMAAN UR RAHMAN, THATIPELLI MANIKANTA, DUGYALA VIVEK, GADDAM VARSHITH, KOTTE NAVEEN, RAGHAVENDRA CHARY** as a Major Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **School of Computer Science and Artificial Intelligence** during the academic year 2024-2025 under our guidance and supervision.

**Mr. Gotte Ranjith Kumar Dr. M.Sheshikala**

Assistant Professor, Professor & Head,

SR University School of CS&AI Anathasagar,Warangal. SR University Ananthasagar, Warangal.

**Reviewer-1 Reviewer-2**

Name: Name:

Designation: Designation:

Signature: Signature:

**ACKNOWLEDGEMENT**

We owe an enormous debt of gratitude toour Capstone project guide **Mr.Ashok Rachapalli, Asst.Prof** as well as Head of the School of CS&AI, **Dr. M.Sheshikala, Professor** andDean of the School of CS&AI, **Dr.Indrajeet Gupta, Professor** for guiding us from the beginning through the end of the Capstone Project with their intellectual advices and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We express our thanks to project co-ordinators **Mr.** **Sallauddin Md, Asst. Prof., and Dr.D.Ramesh Asst. Prof.** for their encouragement and support.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

**ABSTRACT**

The system allows users to control a presentation using hand gestures captured through a webcam. It starts by prompting the user to select a folder containing PNG images representing slides. The program then renames the images sequentially for easier navigation. Once the folder is selected, the webcam captures the user's hand gestures Hand gestures are interpreted to navigate through slides (swipe left/right), annotate slides (draw), erase annotations, and perform additional commands (speech recognition). Speech recognition enables users to issue commands like "open slide X," "next," "previous," "delete," "delete all," and "terminate." These commands facilitate easy navigation and interaction with the presentation. Annotations can be drawn on slides using specific hand gestures, enhancing interactivity during presentations. Additionally, there's an option to delete individual annotations or all annotations on a slide. The system provides visual feedback by overlaying annotations on the slides in real-time. It also displays the total number of slides and a rectangle indicating the area for gesture recognition. In summary, the project combines computer vision, speech recognition, and user interface design to create an intuitive and interactive presentation system.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| S.NO | Content | Page No |
| 1 | Introduction | 1-2 |
| 2 | Related work | 3-5 |
| 3 | Problem Statement | 6 |
| 4 | Requirement Analysis | 7-10 |
| 5 | Risk Analysis | 11-13 |
| 6 | Feasibility Analysis | 14-17 |
| 7 | Proposed approach | 18-20 |
| 8 | Architecture Diagrams | 21-22 |
| 9 | Simulation setup and implementation | 23-32 |
| 10 | Result Comparison and Analysis | 33-38 |
| 11 | Learning Outcome | 39-40 |
| 12 | Conclusion with challenges | 41 |
| 13 | References | 42 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE.NO** | **TITLE** | **PAGE NO** |
| **1** | Class Diagram | 21 |
| **2** | Activity Diagram | 22 |
| **3-8** | Code Implementation | 24-26 |
| **9** | Accuracy Graph | 28 |
| **10** | Confusion Matrix | 29 |
| **11** | Accuracy and Accuracy Val Graph | 30 |
| **12** | Model Workflow | 31 |
| **13** | Sign Up Page | 33 |
| **14** | Sign Up Page | 33 |
| **15** | Home Page | 34 |
| **16** | Uploading brain MRI with Tumor | 34 |
| **17** | Result of Brain MRI-1 | 35 |
| **18** | Uploading brain MRI without Tumor | 35 |
| **19** | Result of Brain MRI-2 | 36 |
| **20** | Uploading lung MRI with Tumor | 36 |
| **21** | Result of lung MRI-1 | 37 |
| **22** | Uploading lung MRI without Tumor | 37 |
| **23** | Result of lung MRI-2 | 38 |
| **24** | History Page | 38 |

**LIST OF ACRONYMS**

KLT Kanade-Lucas-Tomasi

os Operating System

re Regular Expression

OpenCV Open Source Computer Vision Library

numpy Numerical Python

cnn Convolutional Neural Networks

ml Machine Learning

dl Deep Learning

## INTRODUCTION

In our pioneering Interactive Presentation System, we've seamlessly integrated cutting-edge technologies like computer vision, speech recognition, and user interface design. Our primary aim is to redefine the presentation landscape by offering a dynamic and captivating experience where users can effortlessly navigate slides, annotate content, and execute commands with natural hand gestures and voice commands. Through precise hand gesture recognition powered by advanced computer vision algorithms, presenters can interact with their presentations in real time, enhancing engagement and interaction. Additionally, our system leverages state-of-the-art speech recognition capabilities, enabling users to control various aspects of their presentations seamlessly. With a user-centric approach to interface design, we've crafted an intuitive platform that simplifies presentation navigation and interaction, empowering users to deliver compelling and impactful presentations effortlessly.

Built upon a foundation of cutting-edge technologies, our system harnesses the power of the OpenCV library for seamless hand tracking and gesture recognition, allowing users to intuitively navigate slides with simple hand movements. Furthermore, we've seamlessly integrated speech recognition functionalities using the SpeechRecognition library, enabling users to interact effortlessly with the presentation through voice commands. This fusion of computer vision and speech recognition technologies empowers presenters to deliver captivating presentations with enhanced interactivity and engagement. By combining these key components, our system offers a streamlined and immersive presentation experience that elevates communication and audience interaction to new heights.

The user interface is designed to be intuitive and easy to use. Users can select a folder containing their presentation slides using a graphical interface built with the tkinter library. Once the presentation folder is selected, the system automatically renames the slides and prepares them for display.

At the heart of our presentation control system are hand gestures, which serve as the primary method for navigating slides and interacting with content. Users can effortlessly swipe left or right to move between slides, while specific gestures enable annotation and command activation, such as deleting annotations or advancing slides. Powered by the HandDetector module, our system accurately tracks hand movements and interprets gestures in real-time, ensuring seamless and responsive control throughout the presentation. This intuitive gesture-based interface enhances user engagement and simplifies the presentation experience, fostering smoother communication and interaction.

In addition to hand gestures, our system offers voice command functionality, enhancing control and convenience for users. With simple commands like "next slide" or "delete annotation," users can effortlessly navigate through the presentation and manage content using natural language. Leveraging the Google Web Speech API, our system accurately recognizes and processes spoken commands, ensuring smooth and intuitive interaction with the presentation. This integration of voice control adds versatility to the user experience, allowing for hands-free operation and enabling users to focus on delivering their message effectively.

The fusion of gesture-based control, voice recognition, and an intuitive interface results in our Interactive Presentation System providing an immersive and engaging presentation experience. Ideal for various settings such as classrooms, boardrooms, or conferences, this system enables presenters to captivate their audience and deliver impactful presentations effortlessly. With seamless integration and user-friendly features, presenters can navigate slides, annotate content, and execute commands with precision and ease. The system's versatility ensures adaptability to diverse presentation styles and enhances the overall effectiveness of communication and engagement.

## RELATED WORK

In their study, authors Devivara Prasad and Mr. Srinivasulu M from UBDT College of Engineering, India, explore the significance of gesture recognition in Human-Computer Interaction (HCI), emphasizing its practical applications for individuals with hearing impairments and stroke patients. They delve into previous research on hand gestures, investigating image feature extraction tools and AI-based classifiers for 2D and 3D gesture recognition. Their proposed system harnesses machine learning, real-time image processing with Media Pipe, and OpenCV to enable efficient and intuitive presentation control using hand gestures, addressing the challenges of accuracy and robustness. The research focuses on enhancing the user experience, particularly in scenarios where traditional input devices are impractical, highlighting the potential of gesture recognition in HCI.[1]

The paper authored by G. Reethika, P. Anuhya, and M. Bhargavi from JNTU, ECE, Sreenidhi Institute Of Science and Technology, Hyderabad, India, presents a study on Human- Computer Interaction (HCI) with a focus on hand gesture recognition as a natural interaction technique. It explores the significance of real-time hand gesture recognition, particularly in scenarios where traditional input devices are impractical. The methodology involves vision- based techniques that utilize cameras to capture and process hand motions, offering the potential to replace conventional input methods. The paper discusses the advantages and challenges of this approach, such as the computational intensity of image processing and privacy concerns regarding camera usage. Additionally, it highlights the benefits of gesture recognition for applications ranging from controlling computer mouse actions to creating a virtual HCI device.[2]

The paper titled "Smart Presentation Control by Hand Gestures Using Computer Vision and Google’s MediaPipe" was authored by Hajeera Khanum, an M.Tech student, and Dr. Pramod H B, an Associate Professor from the Department of Computer Science Engineering at Rajeev Institute of Technology in Hassan, Karnataka, India. Their research, though lacking a specific publication year, outlines a methodology that harnesses OpenCV and Google's MediaPipe framework to create a presentation control system that interprets hand gestures. Using a webcam, the system captures and translates hand movements into actions such as slide control, drawing on slides, and erasing content, eliminating the need for traditional input devices. While the paper does not explicitly enumerate the challenges encountered during system development, common obstacles in this field may include achieving precise gesture recognition, adapting to varying lighting conditions, and ensuring the system's reliability in real-world usage scenarios. This work contributes to the advancement of human-computer interaction, offering a modern and intuitive approach to controlling presentations through hand gestures.[3]

In their paper titled "Automated Digital Presentation Control Using Hand Gesture Technique," authors Salonee Powar, Shweta Kadam, Sonali Malage, and Priyanka Shingane introduce a system that utilizes artificial intelligence-based hand gesture detection, employing OpenCV and MediaPipe. While the publication year is unspecified, the system allows users to control presentation slides via intuitive hand gestures, eliminating the reliance on conventional input devices like keyboards or mice. The gestures correspond to various actions, including initiating presentations, pausing videos, transitioning between slides, and adjusting volume. This innovative approach enhances the natural interaction between presenters and computers during presentations, demonstrating its potential in educational and corporate settings. Notably, the paper does not explicitly detail the challenges encountered during the system's development, but it makes a valuable contribution to the realm of human- computer interaction by rendering digital presentations more interactive and user-friendly. [4]

The paper titled "A Hand Gesture Based Interactive Presentation System Utilizing Heterogeneous Cameras" authored by Bobo Zeng, Guijin Wang, and Xinggang Lin presents a real-time interactive presentation system that utilizes hand gestures for control. The system integrates a thermal camera for robust human body segmentation, overcoming issues with complex backgrounds and varying illumination from projectors. They propose a fast and robust hand localization algorithm and a dual-step calibration method for mapping interaction regions between the thermal camera and projected content using a web camera. The system has high recognition rates for hand gestures, enhancing the presentation experience. However, the challenges they encountered during development, such as the need for precise calibration and handling hand localization, are not explicitly mentioned in the paper. [5]

The paper "Smart Presentation Using Gesture Recognition" by Meera Paulson, Nathasha P R, Silpa Davis, and Soumya Varma introduces a gesture recognition system for enhancing presentations and enabling remote control of electronic devices through hand gestures. It incorporates ATMEGA 328, Python, Arduino, Gesture Recognition, Zigbee, and wireless transmission. The paper emphasizes the significance of gesture recognition in human- computer interaction, its applicability in various domains, and its flexibility to cater to diverse user needs. The system offers features such as presentation control, home automation, background change, and sign language interpretation. The authors demonstrated a cost- effective prototype with easy installation and extensive wireless signal transmission capabilities. The paper discusses the results, applications, methodology, and challenges, highlighting its potential to improve human machine interaction across different fields.[6]

The paper "Adaptive Hand Gesture Recognition System Using Machine Learning Approach," authored by Rina Damdoo, Kanak Kalyani, and Jignyasa Sanghavi from the Department of Computer Science & Engineering at Shri Ramdeobaba College of Engineering and Management in Nagpur, India, was received on 7th October 2020 and accepted after revision on 28th December 2020. This paper presents a vision-based adaptive hand gesture recognition system employing Convolutional Neural Networks (CNN) for machine learning classification. The study addresses the challenges of recognizing dynamic hand gestures in real time and focuses on the impact of lighting conditions. The authors highlight that the performance of the system significantly depends on lighting conditions, with better results achieved under good lighting. They acknowledge that developing a robust system for real- time dynamic hand gesture recognition, particularly under varying lighting conditions, is a complex task. The paper offers insights into the potential for further improvement and the use of filtering methods to mitigate the effects of poor lighting, contributing to the field of dynamic hand gesture recognition.[7]

This paper, authored by Rutika Bhor, Shweta Chaskar, Shraddha Date, and guided by Prof. M. A. Auti, presents a real-time hand gesture recognition system for efficient human- computer interaction. It allows remote control of PowerPoint presentations through simple gestures, using Histograms of Oriented Gradients and K-Nearest Neighbor classification with around 80% accuracy. The technology extends beyond PowerPoint to potentially control various real-time applications. The paper addresses challenges in creating a reliable gesture recognition system and optimizing lighting conditions. It hints at broader applications, such as media control, without intermediary devices, making it relevant to the human-computer interaction field. References cover related topics like gesture recognition in diverse domains. [8]

In this paper by Thin Thin Htoo and Ommar Win, they introduce a real-time hand gesture recognition system for PowerPoint presentations. The system employs low-complexity algorithms and image processing steps like RGB to HSV conversion, thresholding, and noise removal. It also calculates the center of gravity, detects fingertips, and assigns names to fingers. Users can control PowerPoint presentations using hand gestures for tasks like slide advancement and slideshow control. The system simplifies human-computer interaction by eliminating the need for additional hardware. The paper's approach leverages computer vision and image processing techniques to recognize and map gestures to specific PowerPoint commands. The authors recognize the technology's potential for real-time applications and its significance in human-computer interaction. The references include related works in image processing and hand gesture recognition, enriching the existing knowledge base. [9]

The authors propose a novel method for hands-free control of PowerPoint presentations using real-time hand gestures, eliminating the need for external devices. Their approach involves segmenting the hand in real-time video by detecting skin color, even in varying lighting conditions. The number of active fingers is counted to recognize specific gestures, allowing actions like advancing slides, going back, starting, and exiting the slideshow. The method, implemented with .Net functions and MATLAB, achieved over 90% accuracy in tests with various participants. Challenges include hand positioning variations, potential misplacements, and issues with similar background elements. Future work may focus on accuracy improvement, gesture expansion, and broader software control applications. [10].

## PROBLEM STATEMENT

The problem statement revolves around the development of a hand gesture-controlled presentation application, which aims to modernize traditional presentation methods. This innovative application seeks to replace conventional tools like highlighters and digital pens with intuitive hand gestures, offering presenters greater mobility and interaction capabilities. The envisioned application will allow users to perform essential presentation functions solely through hand gestures, including changing slides, acting as a pointer, writing directly onto slides, and undoing any annotations made. This comprehensive functionality requires accurate gesture recognition and seamless integration with presentation software.To address this challenge, the application will leverage computer vision techniques, including hand tracking and gesture recognition, along with speech recognition for additional commands. The integration of these technologies will enable users to navigate slides, interact with content, and engage audiences effortlessly. The ultimate goal is to redefine the presentation experience by providing a cohesive and intuitive system that enhances interaction dynamics and eliminates the need for traditional tools. With this application, presenters can deliver compelling presentations with ease, fostering greater audience engagement and communication effectiveness

## REQUIREMENT ANALYSIS

**Software Requirements:**

1. Pychram

2. Google Colab

3. VS Code

**Python:**

Ensure that Python is installed on your system. You can download it from the official website

<https://www.python.org/>

**Required Python Packages:**

In this project we need 5 modules in python

1. os - Operating System

2. re - Regular Expression

3. cv2 - OpenCV (Open Source Computer Vision Library)

4. numpy - Numerical Python

5. HandTrackingModule

6. SpeechRecognition Module

**Functional Requirements:**

**a. Speech Recognition:** The system should accurately recognize speech commands spoken by the user using a microphone.

It should support commands such as "next," "previous," "delete," "delete all," "open slide no," and "terminate".

**b. User Interface:**

The system should provide a user-friendly interface for selecting a folder using a file dialog.

It should display slides along with hand-drawn annotations and a webcam feed.

The interface should indicate the total number of slides and the current slide number.

**c. Hand Gesture Recognition:**

Hand gestures should be detected to control slide navigation (e.g., left swipe for previous slide, right swipe for next slide).

Gestures for drawing annotations and erasing content on slides should be recognized.

**d. Slide Control:**

Users should be able to navigate between slides using both speech commands and hand gestures.

The system should allow users to jump to a specific slide by speaking the command "open slide [slide number]."

**e. Annotation and Erasing:**

Users should be able to draw annotations on slides using hand gestures.

There should be support for erasing annotations using specific gestures or commands.

**f. Folder Operations:**

Users should be able to select a folder containing slide images.

The system should rename PNG files in the selected folder with sequential numbers.

**g. Camera Integration:**

Utilize a camera interface to capture hand movements for gesture recognition.

**h. Software Components:**

Develop modules using Python, OpenCV, CV Zone, NumPy, and MediaPipe to execute the hand gesture recognition system effectively.

**Non-functional Requirements:**

**a. Accuracy:**

Ensure a high level of accuracy in recognizing hand gestures to prevent false triggers and ensure seamless presentation control.

**b. Performance:**

Aim for real-time responsiveness in interpreting gestures to maintain a smooth and uninterrupted presentation flow.

**c. Usability:**

Design an intuitive user interface that allows presenters to easily understand and use the hand gestures without complex learning curves.

**d. Compatibility:**

Ensure compatibility with different operating systems and hardware configurations, making the system versatile and accessible.

**e. Reliability:**

Create a robust system that operates consistently across various environmental conditions, lighting situations, and hand orientations.

**f. Security and Privacy:**

Address any potential security concerns related to using a camera interface and ensure user data privacy.

1. **RISK ANALYSIS**

**Technical Challenges:**

**a. Speech Recognition Accuracy:**

**Risk:** Inaccurate recognition due to background noise or unclear speech.

**Mitigation:** Implement noise reduction and train the model with diverse samples.

**b. Hand Gesture Detection:**

**Risk:** Detection accuracy affected by lighting and hand orientation.

**Mitigation:** Use robust algorithms and train with diverse datasets.

**c. Real-Time Performance:**

**Risk:** Performance degradation leading to laggy responses.

**Mitigation:** Profile and optimize critical code sections for speed.

**d. User Interface Complexity:**

**Risk:** Overwhelming interface with many features.

**Mitigation:** Design simple UI, provide clear instructions.

**e. Compatibility Issues:**

**Risk:** Compatibility problems across platforms and devices.

**Mitigation:** Test on multiple platforms, maintain library compatibility.

**User-Related Risks**

**a. User Training and Familiarity:**

**Risk:** Users struggling to understand interaction.

**Mitigation:** Provide user-friendly instructions, gather feedback.

**b. Speech Command Understanding:**

**Risk:** Inaccurate recognition causing confusion.

**Mitigation:** Offer diverse commands, handle variations.

**c. Hand Gesture Learnability:**

**Risk:** Users finding gestures hard to learn.

**Mitigation:** Design intuitive gestures, provide tutorials.

**d. Privacy Concerns:**

**Risk:** User concerns about microphone and camera usage.

**Mitigation:** Communicate privacy policies, offer control options.

**e. Error Handling and Recovery:**

**Risk:** Users encountering errors or unexpected behavior.

**Mitigation:** Implement robust error handling, offer recovery options

## FEASIBILITY ANALYSIS

**Technical Feasibility:**

**a. Technology Stack:** The project utilizes Python, OpenCV, CV Zone, NumPy, and Media Pipe. These technologies are well-established and commonly used for computer vision and machine learning applications, providing robust support.

**b. Hand Gesture Recognition:** OpenCV's capabilities in hand detection and tracking, along with machine learning models, allow the identification and interpretation of hand gestures effectively.

**c. System Architecture:** The system architecture involves modules for hand detection, finger tracking, finger state classification, and gesture recognition, speech recognition which seem technically feasible based on existing libraries and algorithms.

**Operational Feasibility:**

**a.Ease of Use:** The proposed system aims to simplify the presentation process by allowing users to control slides using hand gestures. This can potentially make presentations more intuitive and engaging for both presenters and audiences.

**b.Compatibility:** The system's compatibility with different presentation formats (e.g., PowerPoint, images) needs to be considered for seamless integration with various presentation tools.

1. **PROPOSED APPROACH**

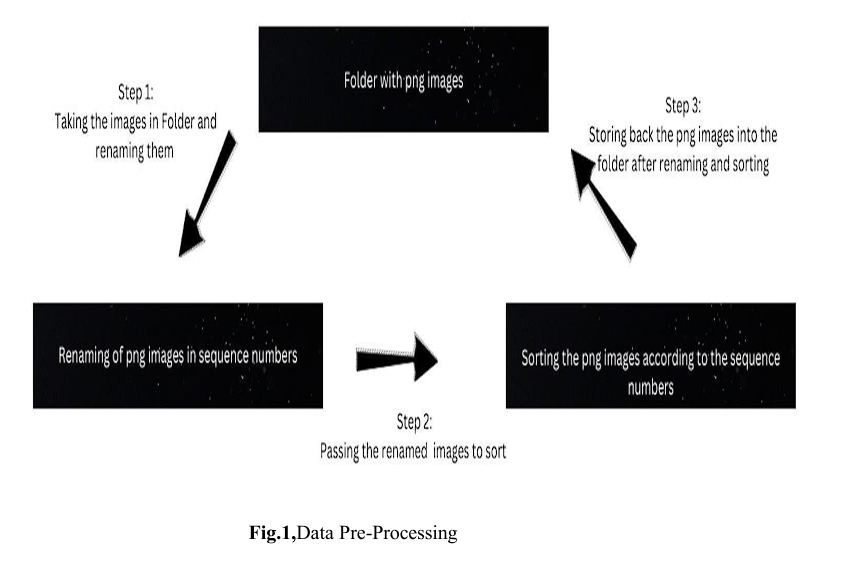
**1. Data Collection and Preprocessing:**

**Input:** PNG Images of PowerPoint slides

**Process:**

**Data Collection :** Gather PowerPoint slides and convert them into PNG image format.

**Data Preprocessing:** Organize the PNG images in a folder in sequential order for easy retrieval and usage during presentations.



**2. Building the Hand Gesture Recognition and Speech Recognition Model:** **Programming Language:** Python

Libraries:

**Os:** Operating System.

**Re:** Regular Expression.

**OpenCV (cv2):** For computer vision tasks, including image processing, detection, and tracking.

**NumPy:** For numerical operations and array manipulations.

**Custom HandTrackingModule:** A module for hand detection, finger tracking, and gesture recognition. Likely built using machine learning or custom algorithms.

**Speech Recognition:** A module in Python helps programs understand and process spoken language. It listens to audio input, like what you say into a microphone, and converts it into text that the program can understand and work with.

**3. Hand Gesture Detection and Interpretation:**

**Hand Detection:** Utilize computer vision techniques to detect the presence and location of hands in the video frame.

**Finger Tracking:** Track individual fingers' positions and movements within the hand. **Gesture Recognition:** Identify and classify specific hand gestures based on finger positions and movements.

**Gesture-to-Action Mapping:** Define actions such as next slide, previous slide, write/draw, delete, pointer, and exit/terminate presentation corresponding to recognized gestures.

**4.Speech Recognition and Interpretation:**

**Audio Capture:** Capture audio input, either from a microphone or an audio file. Preprocessing (Optional): Perform preprocessing tasks like adjusting for ambient noise or enhancing audio quality.

**Speech Recognition:** Utilize the speech recognition functionality to convert the captured audio into text.

**Interpretation:** Interpret the recognized text to derive meaning or context relevant to the application.

**Action Mapping:** Map the recognized speech to specific actions or commands based on predefined criteria.

**Error Handling:** Implement error handling to manage situations where the speech cannot be accurately recognized or understood.

**5.Implementation and Testing:**

**User Interaction:**

Allow users to interact with the presentation using predefined hand gestures.

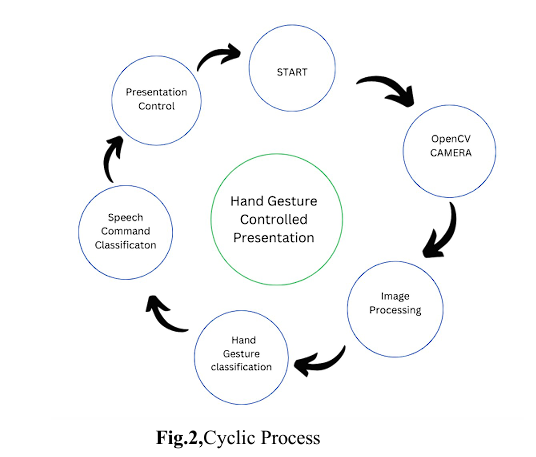
Utilize hand gesture recognition to control slide transitions, writing, erasing, highlighting, and other actions.

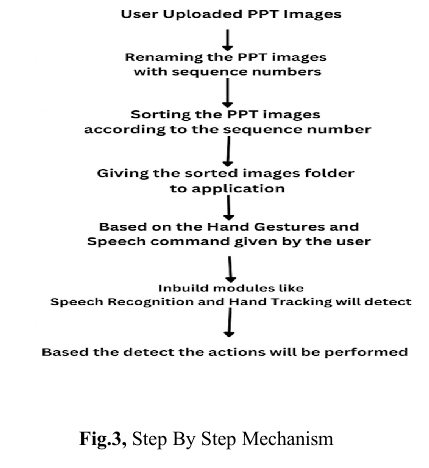
**Presentation Control:** Trigger actions such as next slide, previous slide, write/draw, delete, pointer, and exit/terminate presentation based on recognized gestures. Implement speech recognition to provide an alternative means of user interaction, allowing users to control the presentation using voice commands.

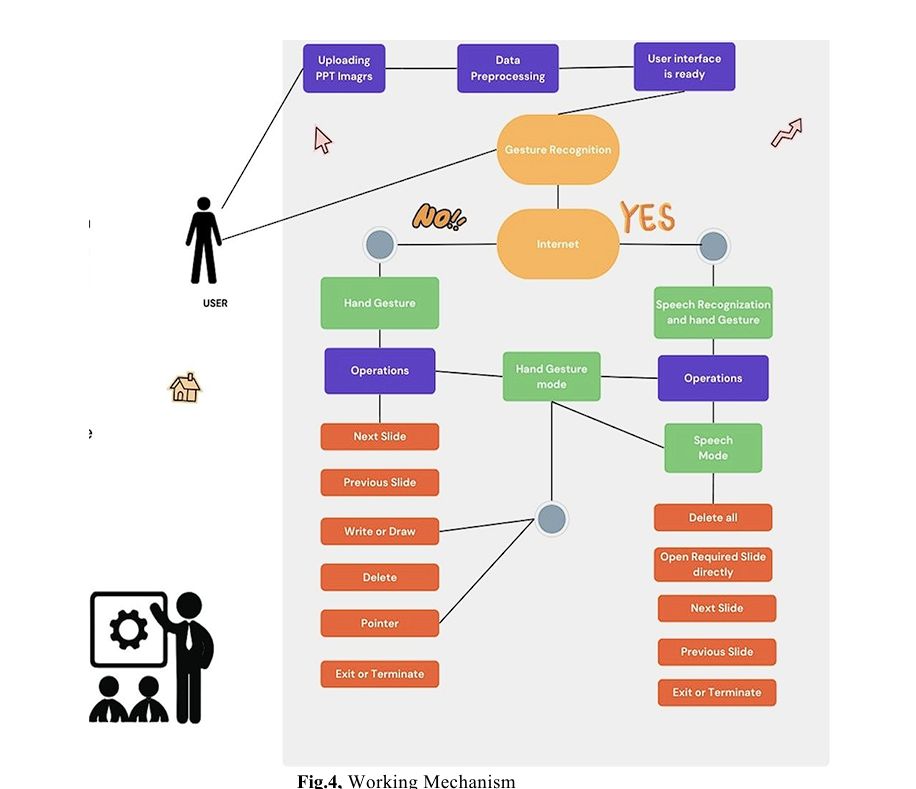
**Testing:** Test the system extensively with various hand gestures and speech commands to ensure accurate recognition and reliable action execution.

Conduct usability testing to gather feedback from users and refine the system based on their experience and suggestions.

1. **ARCHITECTURE DIAGRAMS**





****

### SIMULATION SETUP AND IMPLEMENTATION

**9.1. Simulation setup**

**Hardware Requirements:**

Hardware : intel core i5/i7

Speed : 2.5 GHz

RAM : 8GB

Web camera : HD (720p) resolution

Microphone : Good Frequency Response Covers the human voice(around 80 Hz to 14 kHz).

**Software Requirements:**

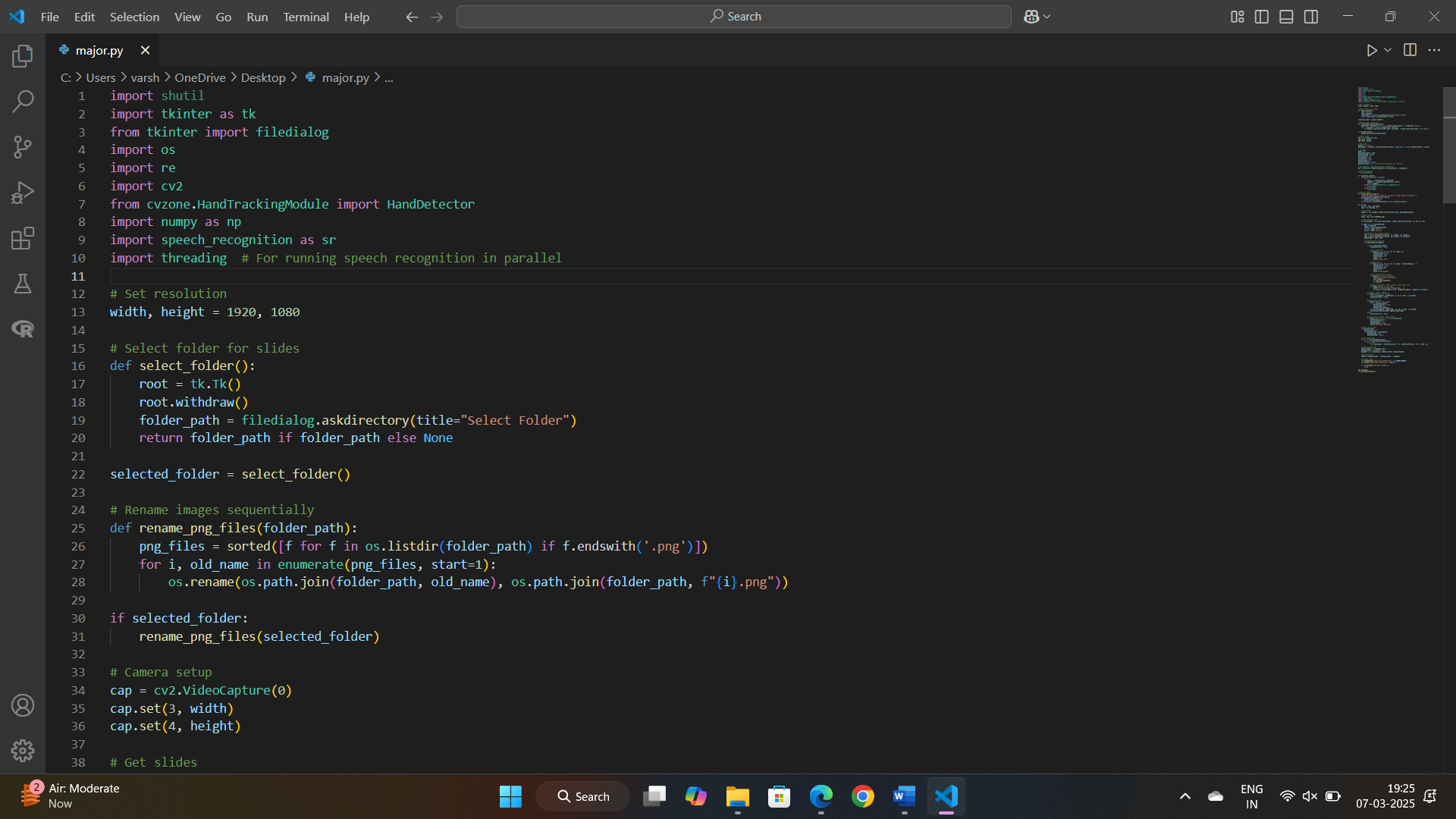
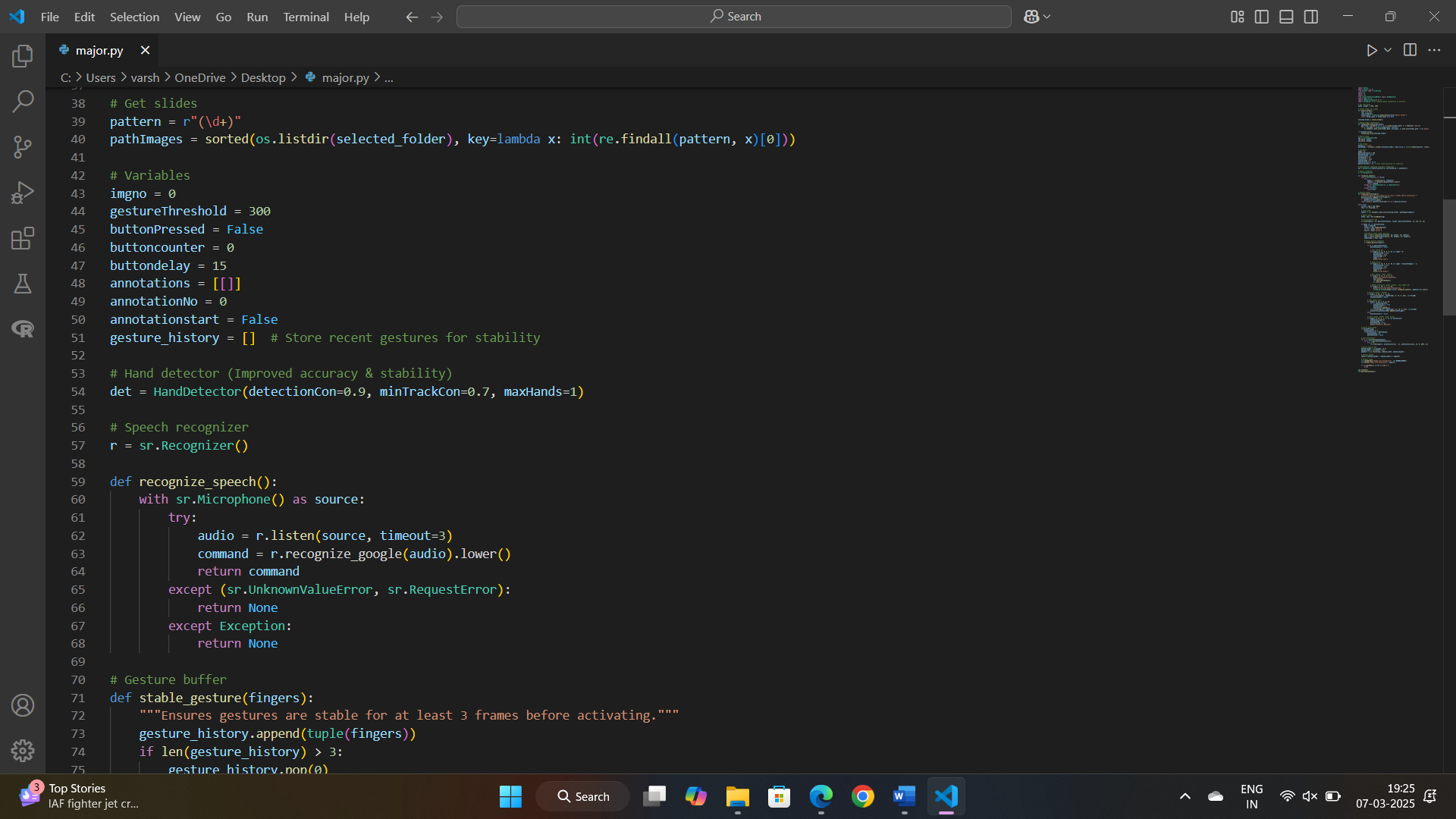
Operating System : Windows/Mac OS

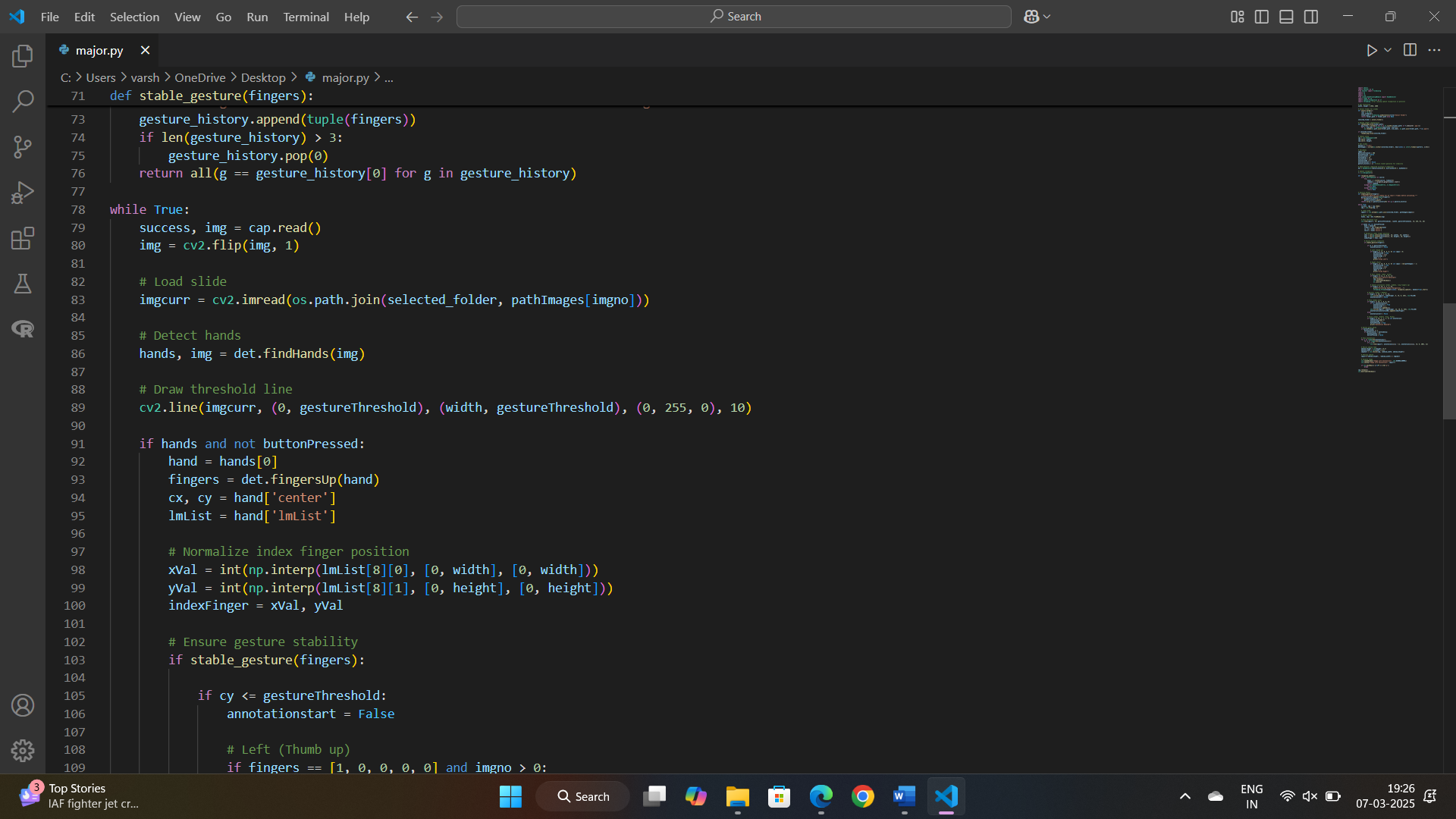
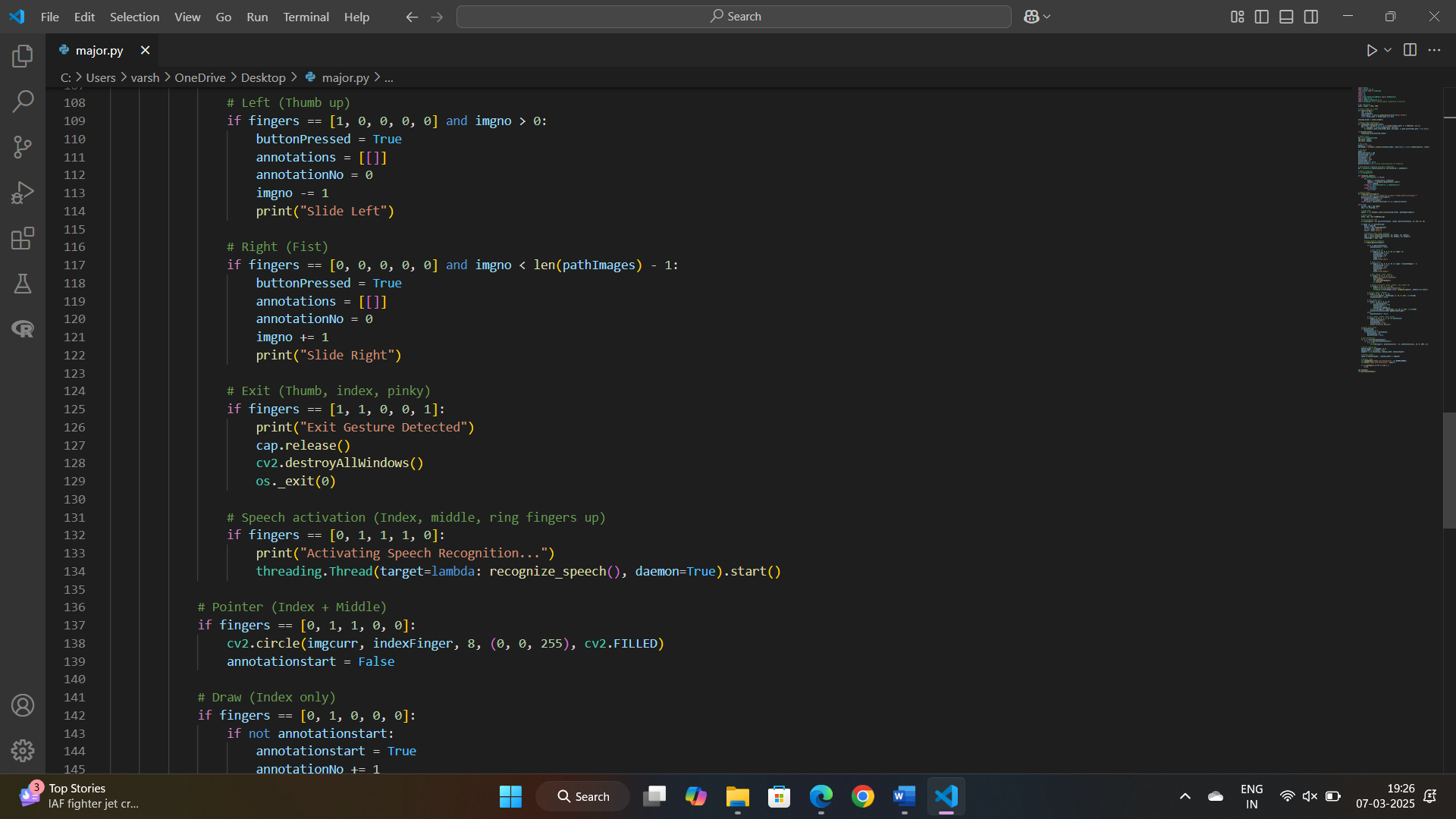
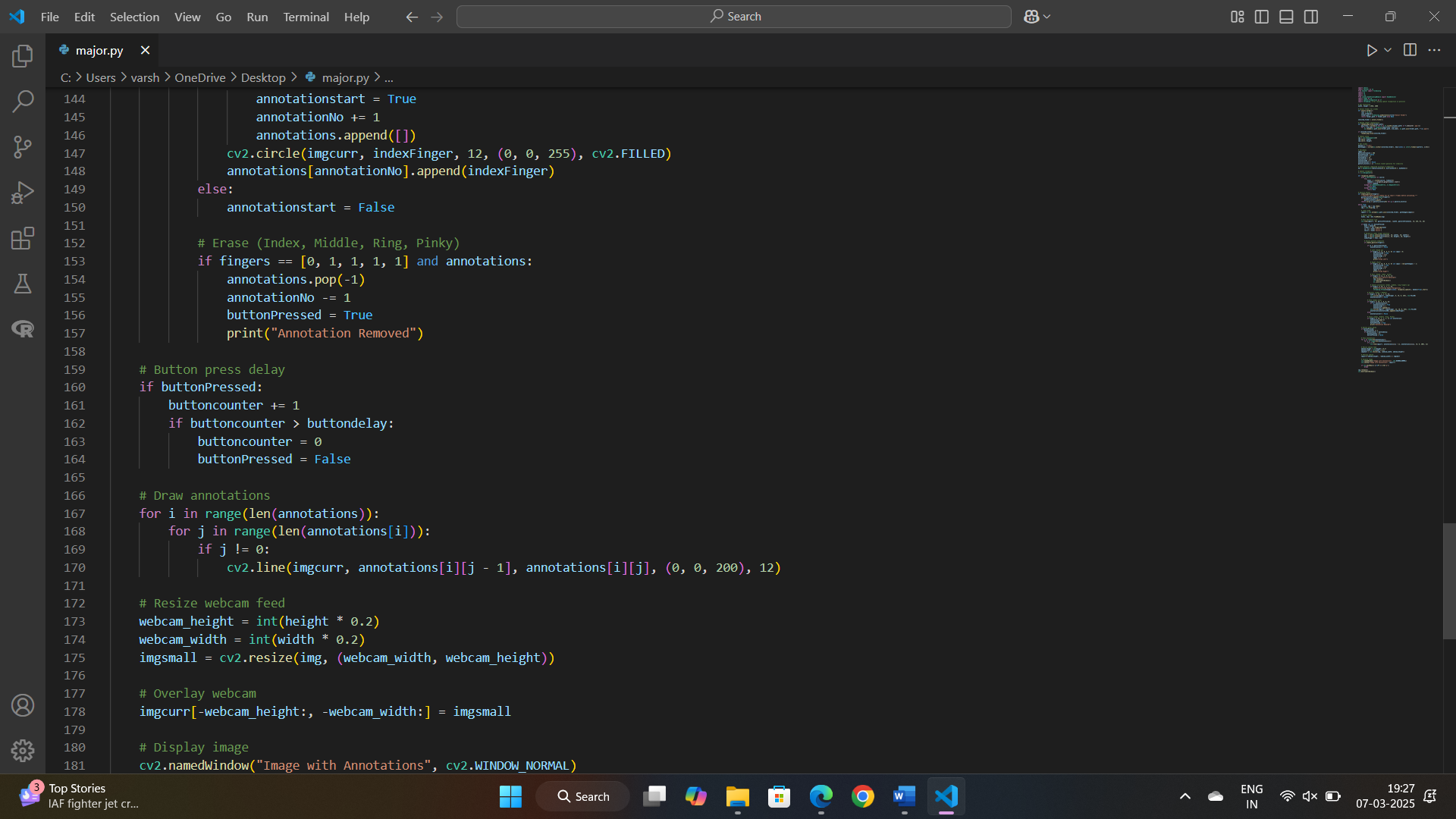
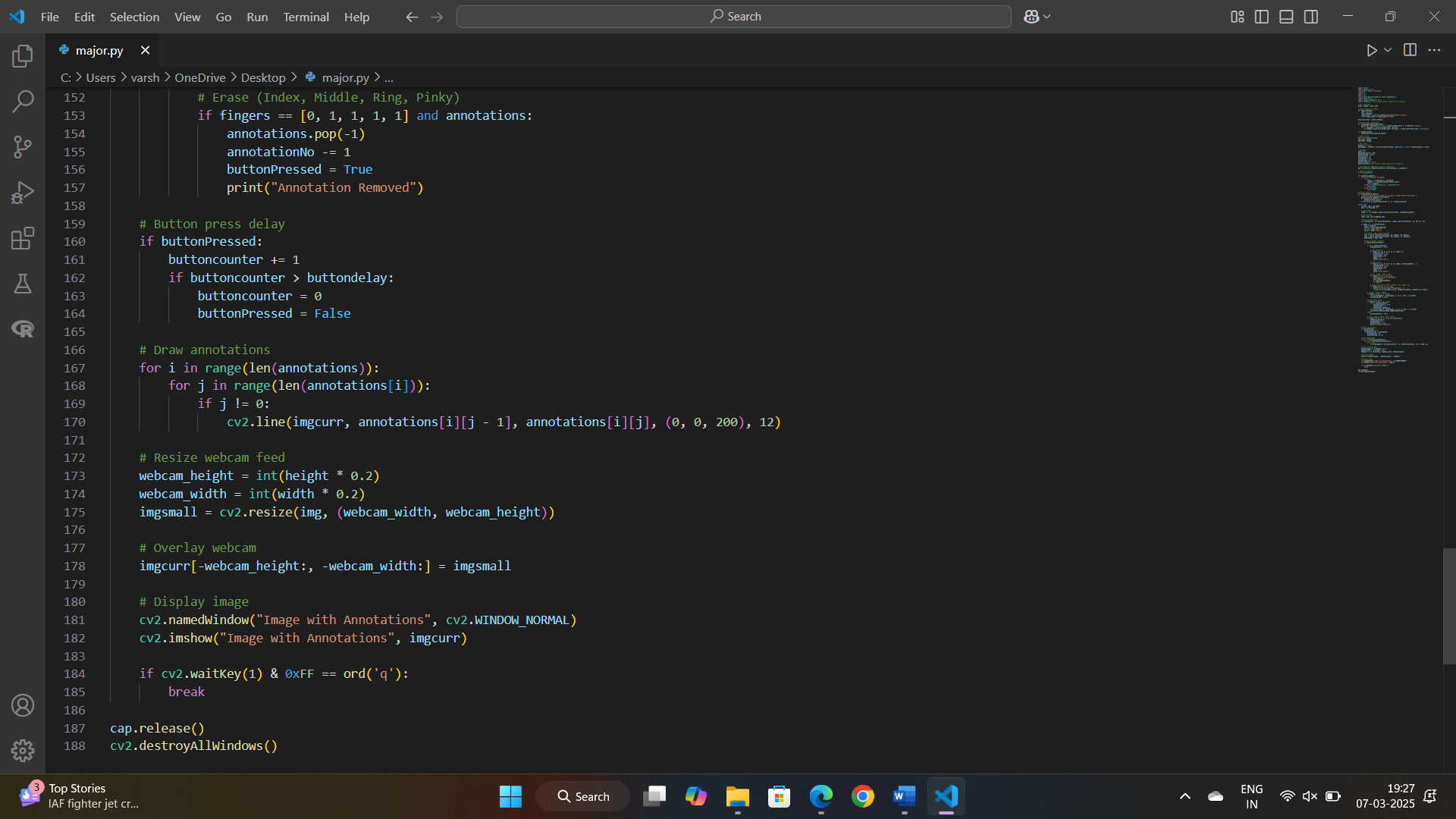
Technology : Deep Learning

Platform : Visual Studio Code, Pycharm CE

Libraries : Os, re, OpenCV, HandTrackingModule, Speech recognition, numpy

**9.2. Implementation**

 ****

**Fig 3,** Code Implementation     **Fig 4,** Code Implementation

**1. User Uploaded PPT Images:**

Users upload PowerPoint slides which are converted to PNG format.

**2. Renaming the PPT Images with Sequence Numbers:** The system assigns sequential numbers to each uploaded PNG image to ensure the slides are in a specific order.

**3. Sorting the PPT Images:** After renaming, the images are sorted based on the assigned numerical sequence to ensure the correct order of slides.

**4. Storing the Images Folder to Variables:** The folder containing the sorted PNG images is stored as a variable or accessed for further processing in the Python environment.

**5. Importing Hand Tracking Module (KLT Algorithm):** A custom Hand Tracking Module utilizing the Kanade-Lucas-Tomasi (KLT) algorithm is imported. This module enables hand detection, finger tracking, and gesture recognition.

**6. Gesture Recognition using OpenCV Camera & KLT Algorithm:** The system utilizes OpenCV's camera functionality to capture real-time video frames.

The KLT algorithm within the Hand Tracking Module is employed to detect and track the position of the user's hand within the video frame.

**7. Gesture-Based Actions on PPT Slides:**

Recognition of Specific Hand Gestures: The system recognizes predefined hand gestures using the tracked hand positions and movements obtained through the KLT algorithm.

Mapping Gestures to Actions: Each recognized gesture is mapped to a specific action related to controlling the PowerPoint slides. For example:

Gestures such as swiping left or right can trigger slide transitions (next or previous). A pointing gesture can function as a pointer on the slides.

Specific finger configurations or movements can be mapped to actions like writing, erasing, or highlighting content on the slides.

A certain gesture can be designated to exit or terminate the presentation.

**8.Initialization and Configuration:**

The speech recognition module (speech\_recognition) is imported at the beginning of the code. An instance of the Recognizer class is created and assigned to the variable recognizer. The recognize\_speech() function is defined to handle speech recognition tasks.

**9.Speech Recognition Function (recognize\_speech()):**

This function uses the microphone as the audio source to listen for user commands. When invoked, it displays a "Listening..." message on the image combined with the webcam feed (img\_combined) to indicate that it's ready to receive commands.

It adjusts for ambient noise for one second using recognizer.adjust\_for\_ambient\_noise() to improve recognition accuracy.

The listen() method of the recognizer records audio input from the microphone until a pause or silence is detected.

The recorded audio is then passed to the Google Web Speech API (recognize\_google()) for speech recognition.

If the API successfully recognizes the speech, the recognized text is printed, and the function returns the recognized text in lowercase.

If the API fails to recognize the speech due to unknown input or connection issues, appropriate error messages are displayed, and an empty string is returned.

**10.Integration with Gesture Recognition:**

Within the main loop of the code, after hand detection and gesture recognition, a condition checks if all fingers are raised (fingers == [1, 1, 1, 1, 1]), indicating a closed fist or "fist bump" gesture. Upon detecting this gesture, the recognize\_speech() function is called to listen for the user's voice command. Based on the recognized text, specific actions are performed, such as navigating to the next or previous slide, opening a specific slide, deleting annotations, terminating the presentation, or deleting all slides. The recognized text is used to trigger actions within the presentation control logic, providing a seamless integration of voice commands with hand gestures for controlling PowerPoint slides.

**Kanade-Lucas-Tomasi (KLT) algorithm**

**1. Preprocessing:**

Begin by capturing an initial frame from a video feed or image sequence that includes the hand(s) you want to detect. Convert the frame to a suitable color space (like grayscale) to simplify subsequent computations.

**2. Feature Detection:**

Apply a feature detection method (e.g., Harris corner detection, FAST features, etc.) to identify distinctive points or corners within the image that can represent potential features of the hand.

**3. Feature Tracking Initialization:**

Select the features that are within the region(s) of the hand in the initial frame. These features serve as the starting point for tracking the hand's movement across subsequent frames.

**4. Tracking the Features:**

For each feature detected and selected, track its movement in the subsequent frames of the video sequence using the KLT algorithm. The KLT algorithm tracks the movement by estimating the optical flow, i.e., how the pixels or features move between frames. It does so by finding the best matching points between consecutive frames.

**5. Updating Feature Set:**

As the frames progress, some features might get occluded or become unreliable for tracking due to factors like lighting changes or hand movement. Constantly update and reinitialize the feature set by detecting new features in the regions where the hand is expected to be present.

**6. Hand Region Estimation:**

Aggregate the tracked features that consistently represent the hand across multiple frames. Using geometric or statistical methods (like bounding box estimation around the tracked features), define the region where the hand is detected.

**7. Hand Gesture Recognition (Optional):**

After detecting and tracking the hand region, additional algorithms or machine learning models can be employed to recognize specific gestures or actions performed by the hand.

**8. Feedback and Refinement:** Evaluate the accuracy of the hand detection and refine the process if needed by adjusting parameters, selecting better features, or employing more sophisticated algorithms.

The KLT algorithm, when adapted for hand detection, provides a framework for continuously tracking features representing the hand across video frames. This allows for real-time estimation of hand movement and location, enabling applications in gesture recognition, human-computer interaction, and more. However, it's essential to consider potential challenges like occlusions, lighting variations, and variations in hand appearance for robust hand detection using this approach.

**Speech Recognition Engines or APIs:**

**1.Audio Capture and Preprocessing:**

The module captures audio input from the microphone as an audio stream. It may perform preprocessing steps like noise reduction or normalization to enhance the quality of the audio input.

**2.Feature Extraction:** The audio stream is converted into a format suitable for analysis, often in the form of spectrograms or other feature representations. This step extracts relevant features from the audio signal that can be used for recognition.

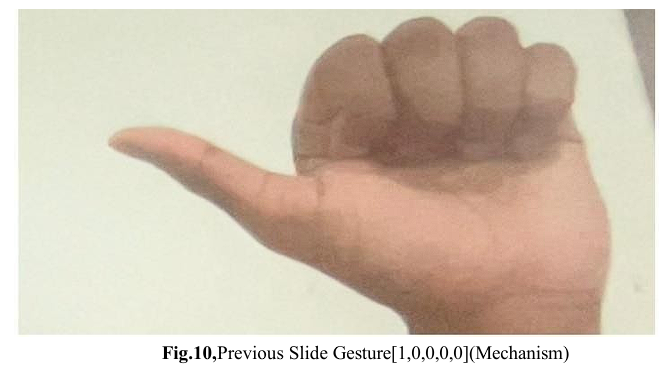
**3.Speech Recognition Engine:** The module interfaces with various speech recognition engines or APIs, such as Google Web Speech API, CMU Sphinx, IBM Watson Speech to Text, etc. Each recognition engine utilizes its own algorithms and models for transcribing speech. These engines typically employ sophisticated machine learning techniques, including deep learning models like convolutional neural networks (CNNs) or recurrent neural networks (RNNs), along with language models and acoustic models.

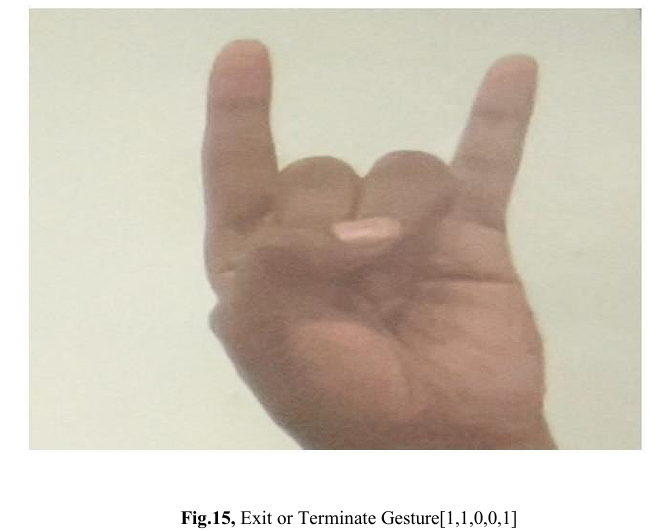
**4.Acoustic and Language Models:** Acoustic models map audio features to phonemes or basic speech sounds. Language models incorporate linguistic knowledge to predict word sequences and improve recognition accuracy. Some engines may use statistical methods, while others rely on neural network-based approaches.

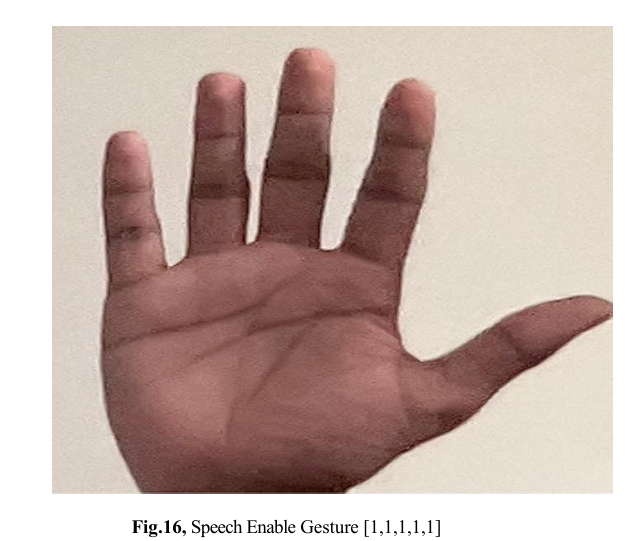
**5.Recognition and Decoding:** The recognition engine processes the audio input and generates a transcription hypothesis, i.e., a sequence of words that best matches the input audio. This involves decoding the audio features using the acoustic and language models to estimate the most likely sequence of words. The engine may employ techniques like Hidden Markov Models (HMMs), Connectionist Temporal Classification (CTC), or sequence-to-sequence models for this task.

**6.Output:** The recognized text output is returned to the calling program for further processing or display. The module may also provide additional metadata, such as confidence scores or timing information, to assess the accuracy and reliability of the transcription.

**Gesture Mechanism:**







Speech Enable Gesture will switch to the speech recognition mode where we can give commands to the program through voice

It will enable operations like

1. Next Slide command as “next”.

2. Previous Slide command as “previous”.

3. Delete command as “delete”.

4. Delete All command as “delete all”.

5.Terminate the presentation command as “terminate”.

6.Open Any Slide command as “open slide No”.

7. only Writing on the slide will be accessed by hand gesture.

Speech Enable Gesture contain 2 major works

1. Listening Mode (Listen to the user).

2. Recognizing Mode (Recognizing what user send).

The use of a green line on the screen serves as a visual guide and segmentation element for users during a presentation, particularly for gesture-based interactions. By dividing the screen with this line, it delineates specific regions for different gesture functionalities, enhancing the user experience and ensuring accuracy in gesture recognition. The allocation of gestures above and both above and below the line is purposeful, catering to different functionalities based on their positioning relative to the line. **Gestures Above the Line:**

**Previous Slide Gesture, Next Slide Gesture, Exit Gesture, Speech Enable Gesture :**

These gestures are exclusively designed to work in the space above the green line. They facilitate essential presentation controls, such as moving to the previous or next slide and exiting the presentation mode and switching to speech recognition mode. Restricting these gestures to the area above the line ensures a clear and distinct control space for navigating through slides without interference from other functionalities.

**Gestures Above and Below the Line:**

**Pointer Gesture:**

This gesture allows users to activate a pointer function, enabling them to interact with content or highlight specific areas on the slides. By allowing this gesture both above and below the line, users can seamlessly control the pointer regardless of its position relative to the line.

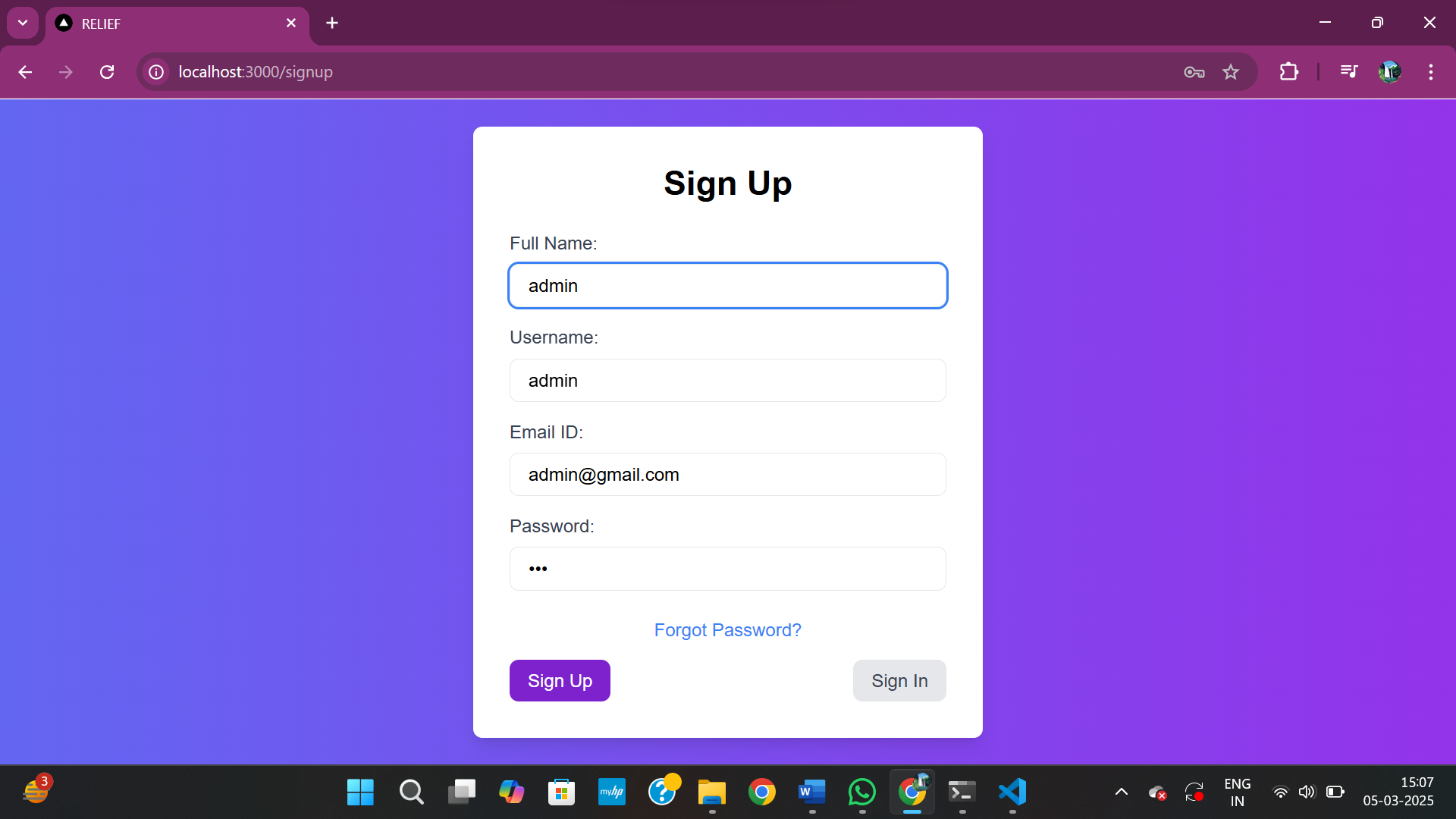
**Write & Draw Gesture:**

Enabling this gesture both above and below the line grants users the ability to annotate or draw on the slides. It offers flexibility for users to create annotations wherever they find it comfortable, whether it's above or below the green line.

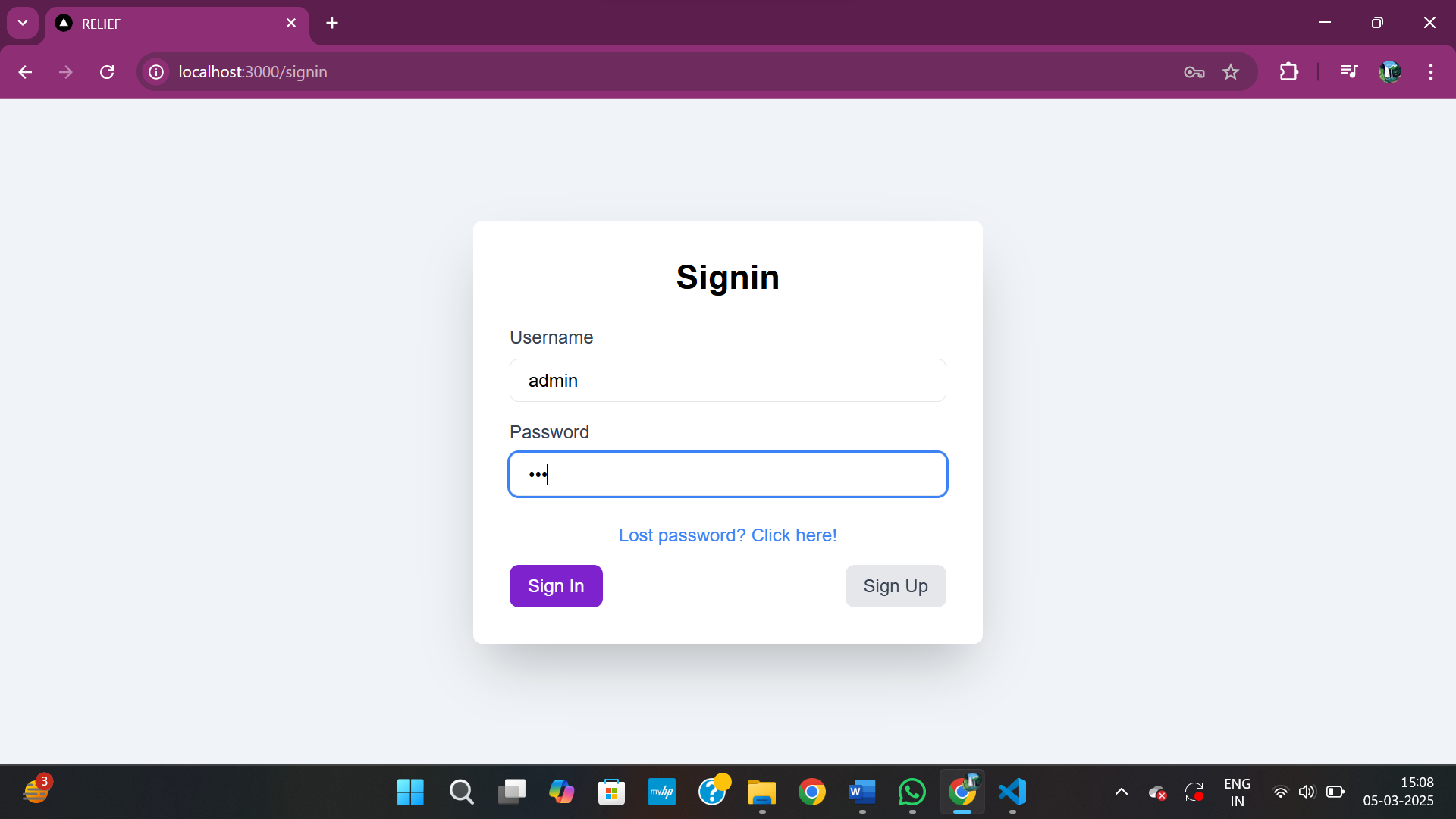
**Delete Gesture:**

The delete gesture is also made accessible in both regions to provide users the capability to erase or delete any annotations or drawings made on the slides. This ensures ease of interaction and correction, regardless of the position of the drawn content in relation to the green line. The visual cue of the green line offers a clear demarcation for users, simplifying the navigation of various presentation functionalities using hand gestures. It optimizes user control and minimizes confusion, ensuring a smooth and intuitive interaction with the presentation content while offering distinct areas for specific gesture-based action

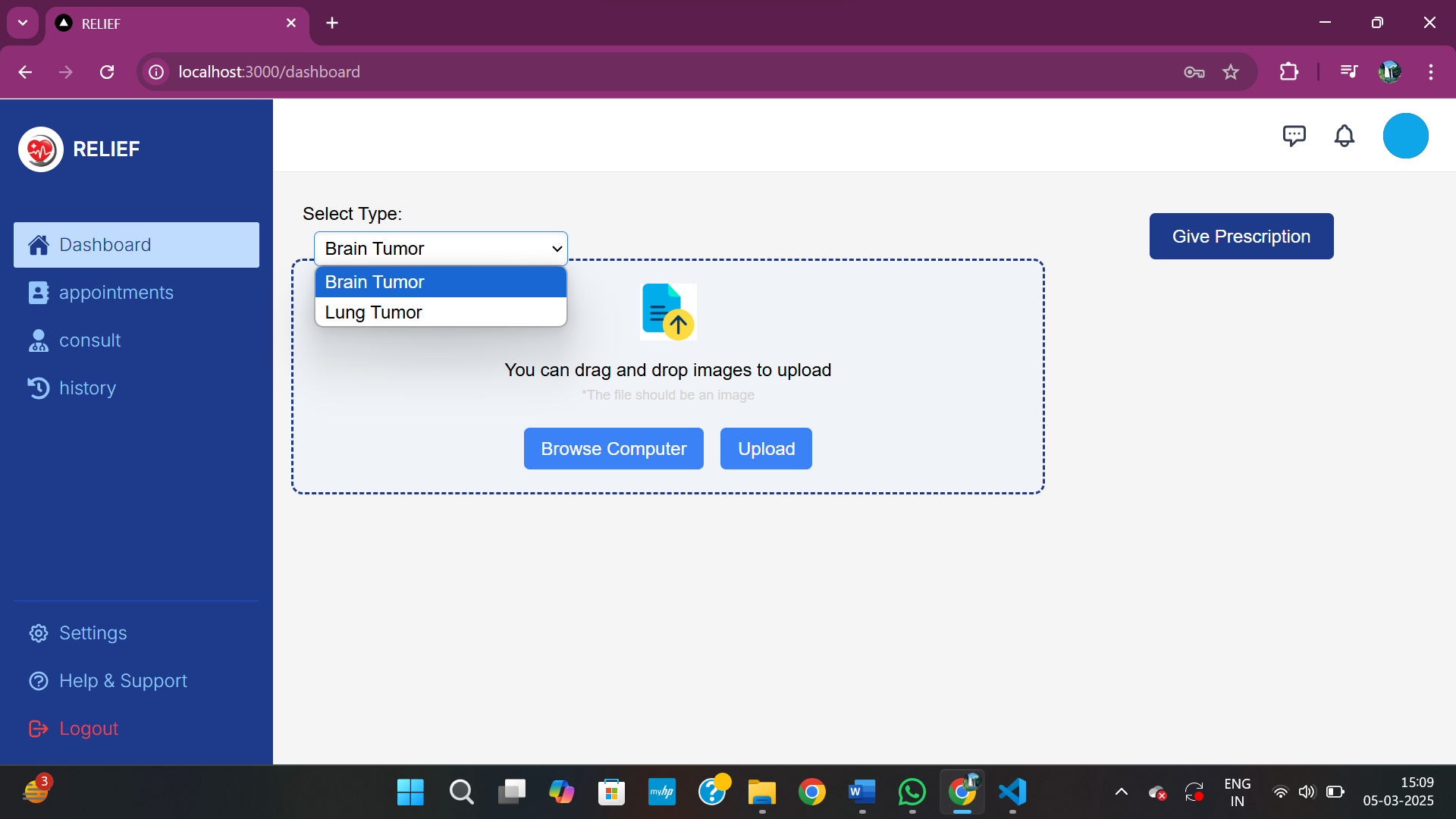
**10.RESULT COMPARISION AND ANALYSIS**



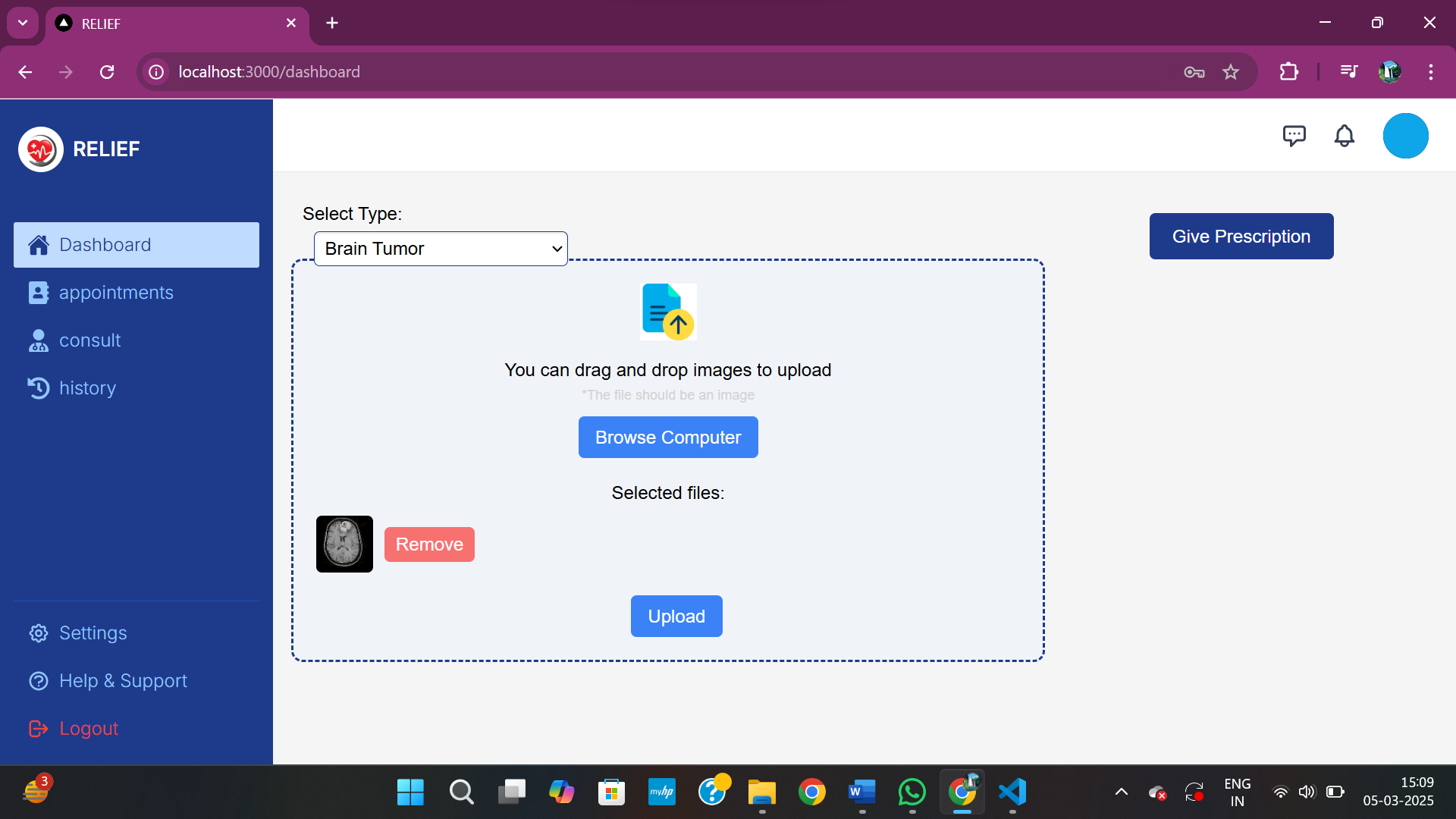
**Fig 13,** Sign Up Page



**Fig 14,** Sign In Page



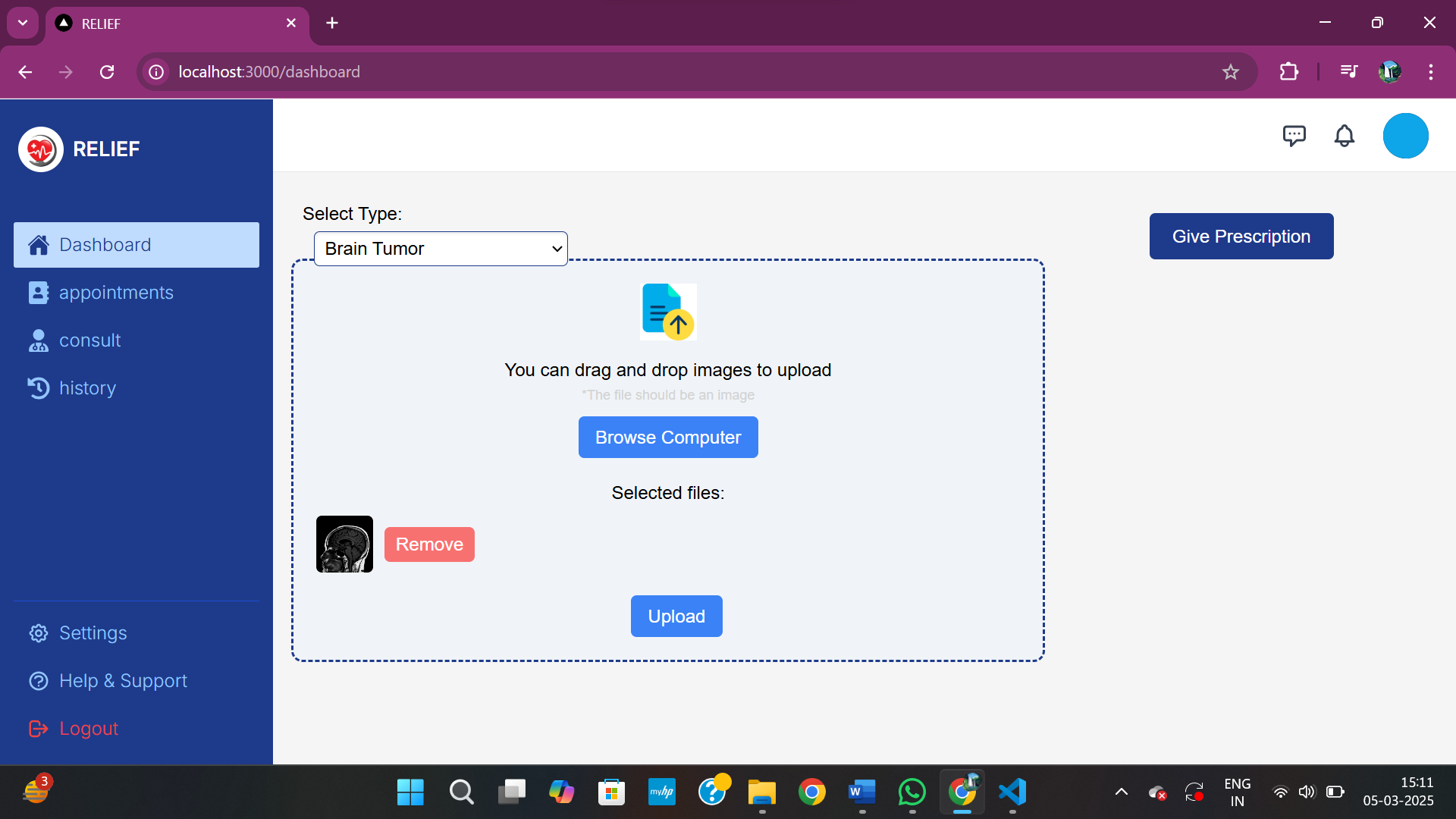
**Fig 15,** Home Page



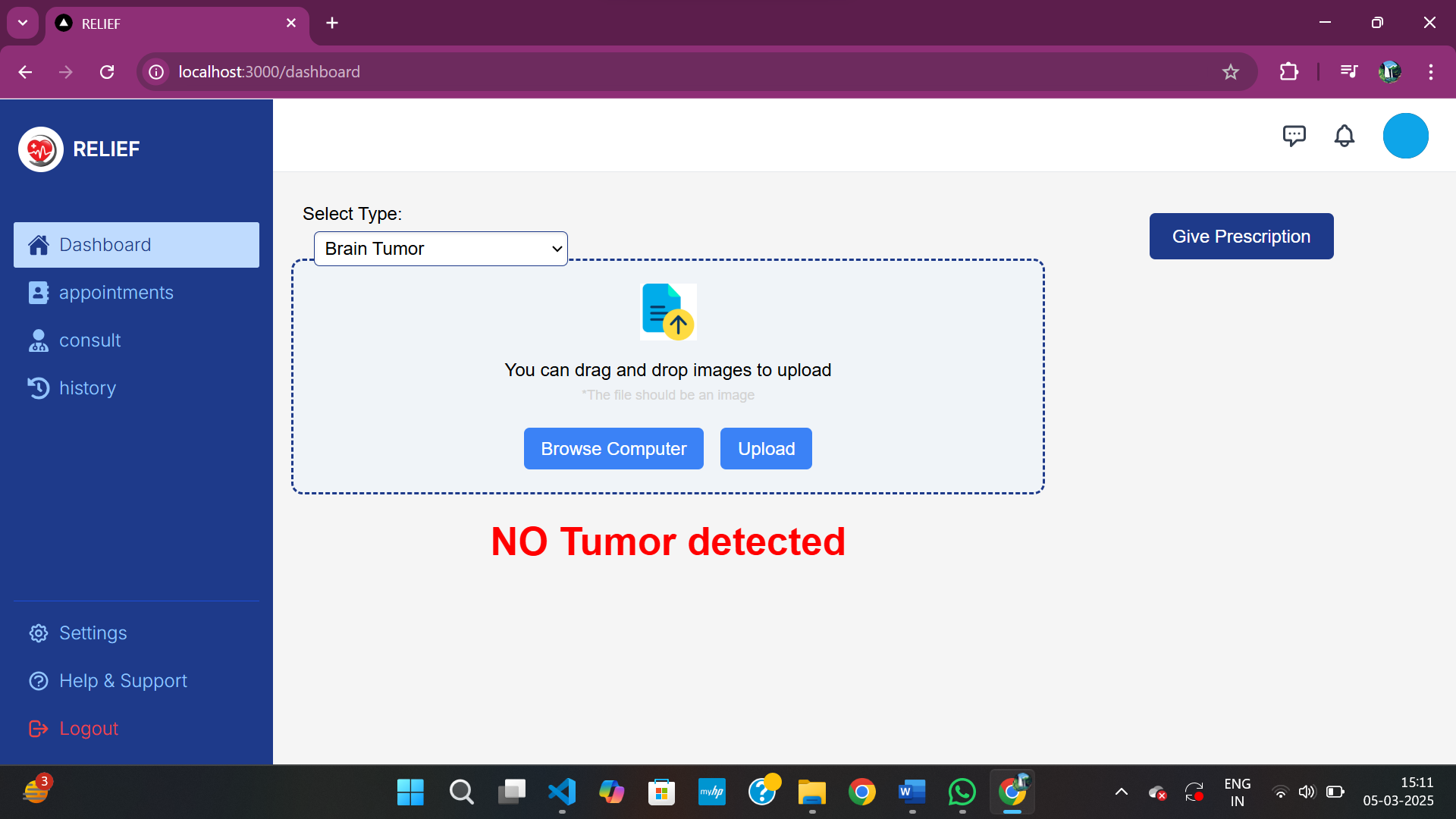
**Fig 16,** Uploading brain MRI with Tumor

### 

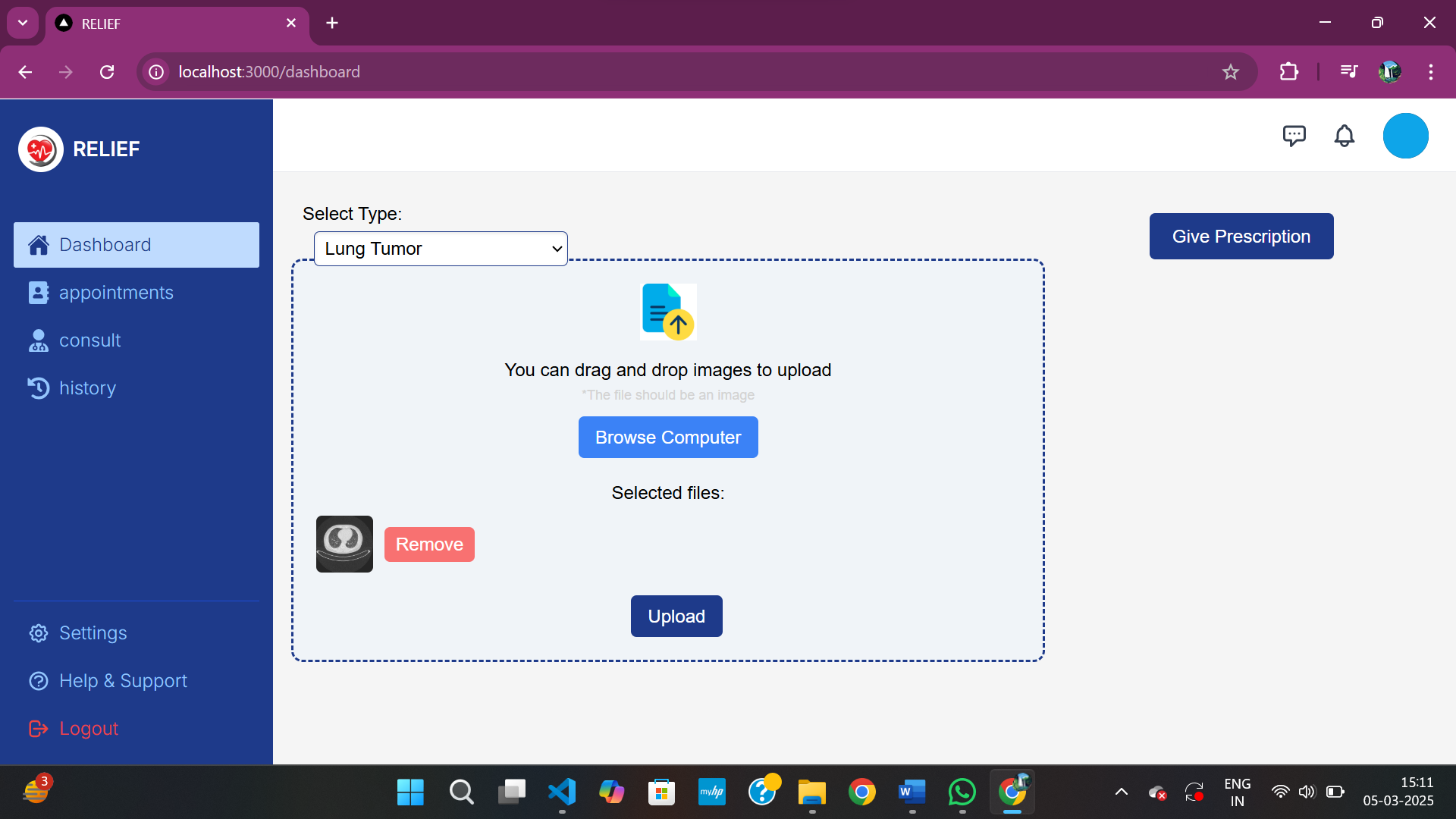
**Fig 17,** Result of Brain MRI-1



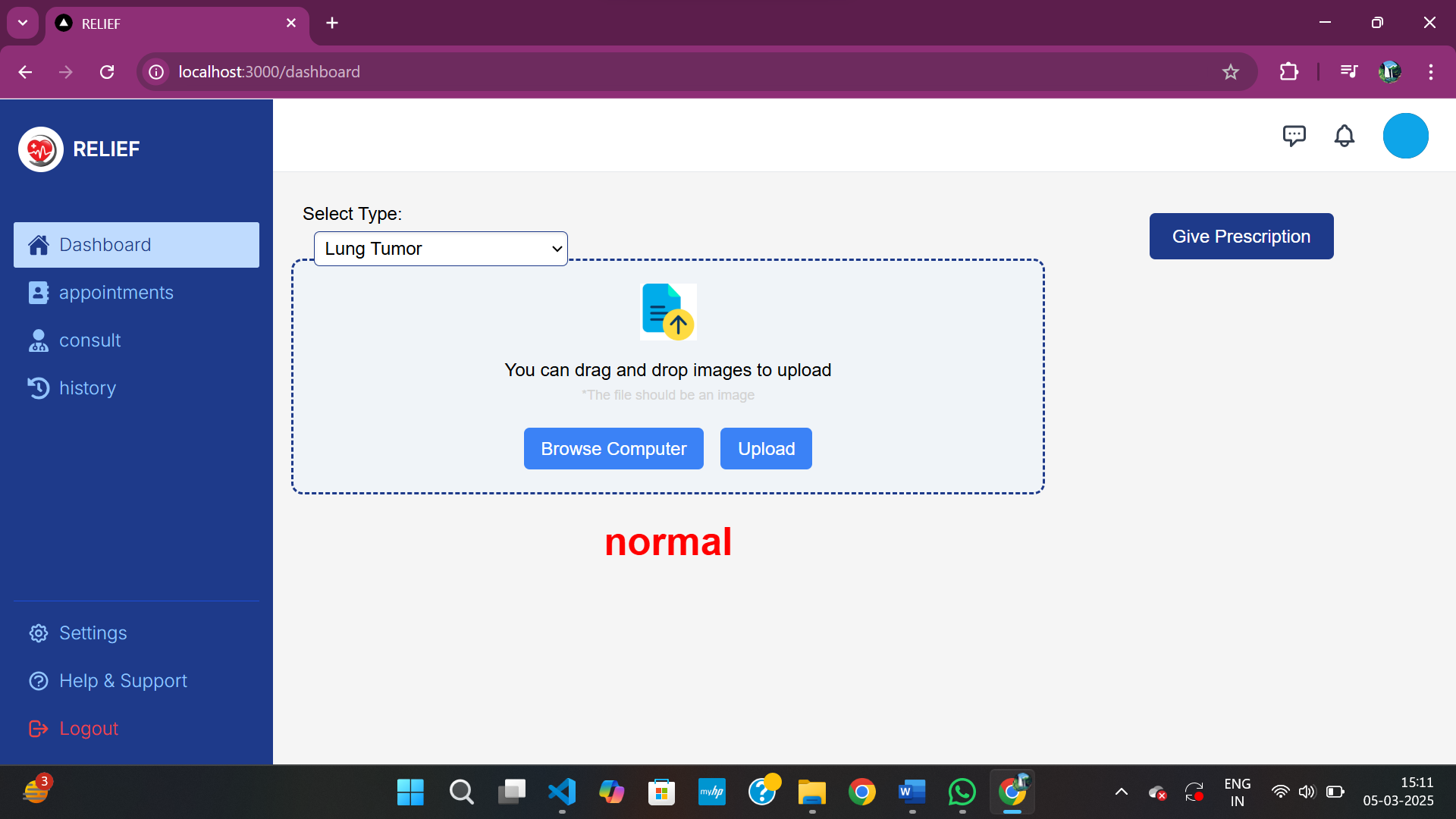
**Fig 18,** Uploading Brain MRI without Tumor



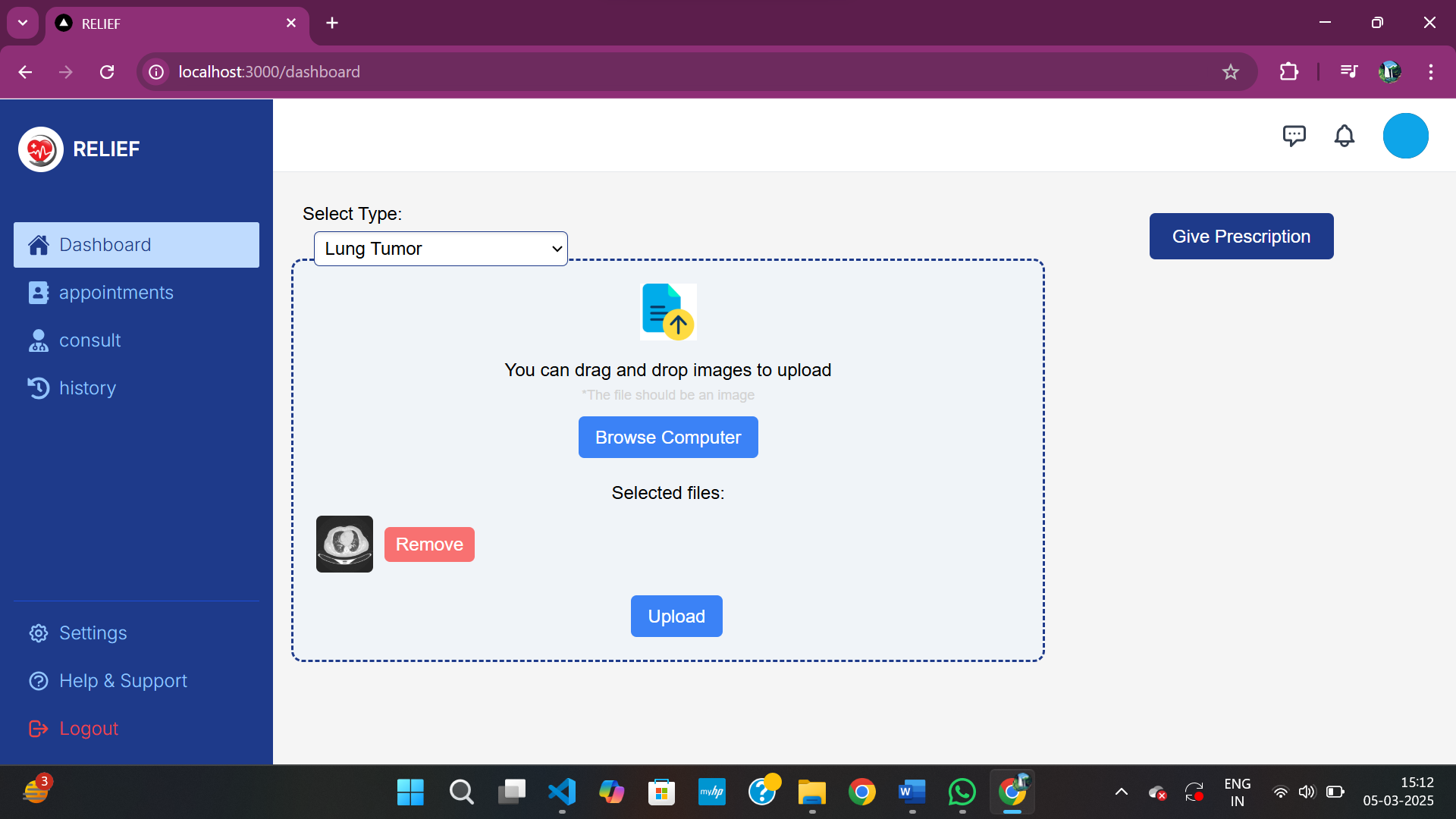
**Fig 19,** Result of Brain MRI-2



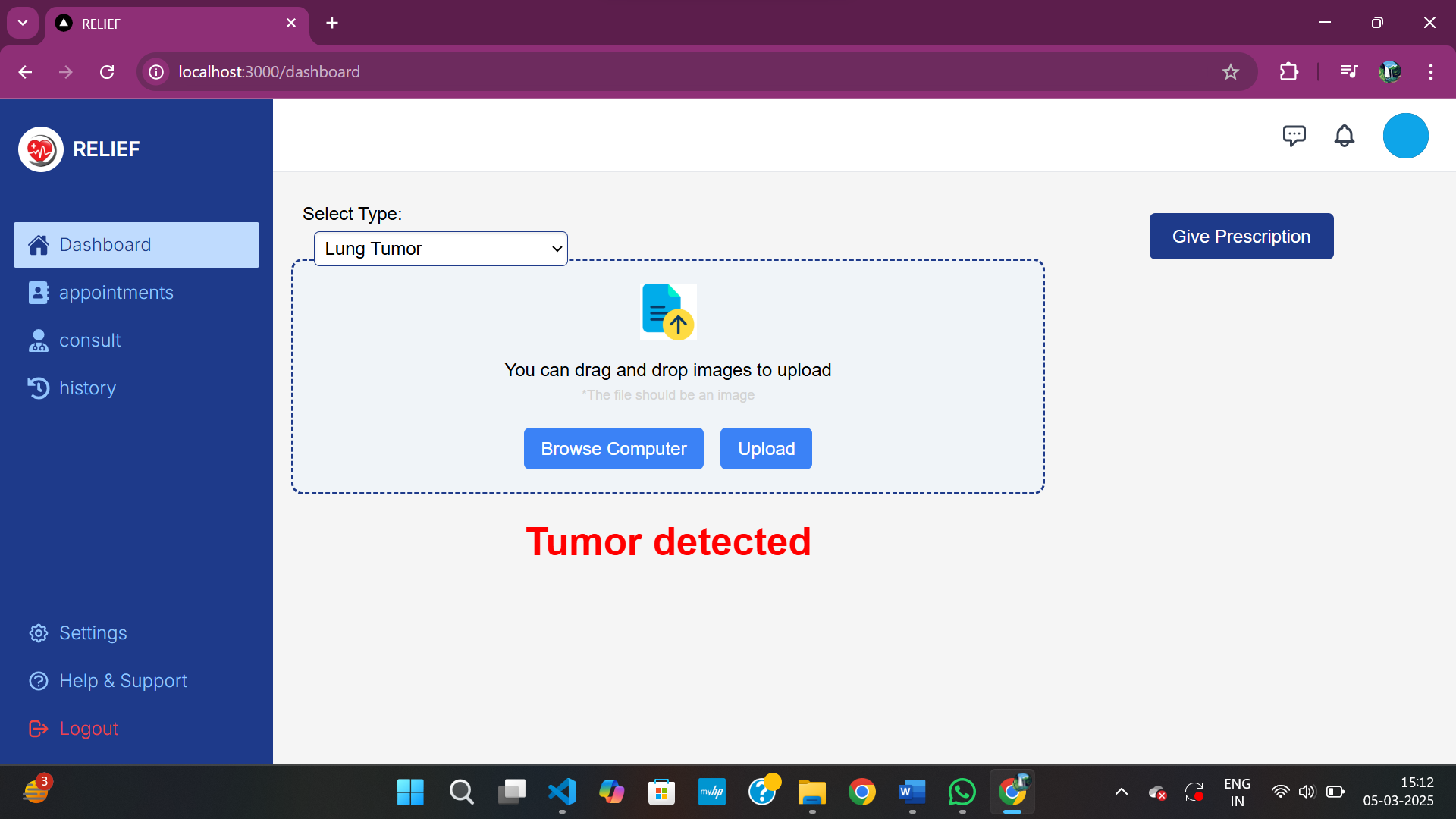
**Fig 20,** Uploading Lung MRI without Tumor



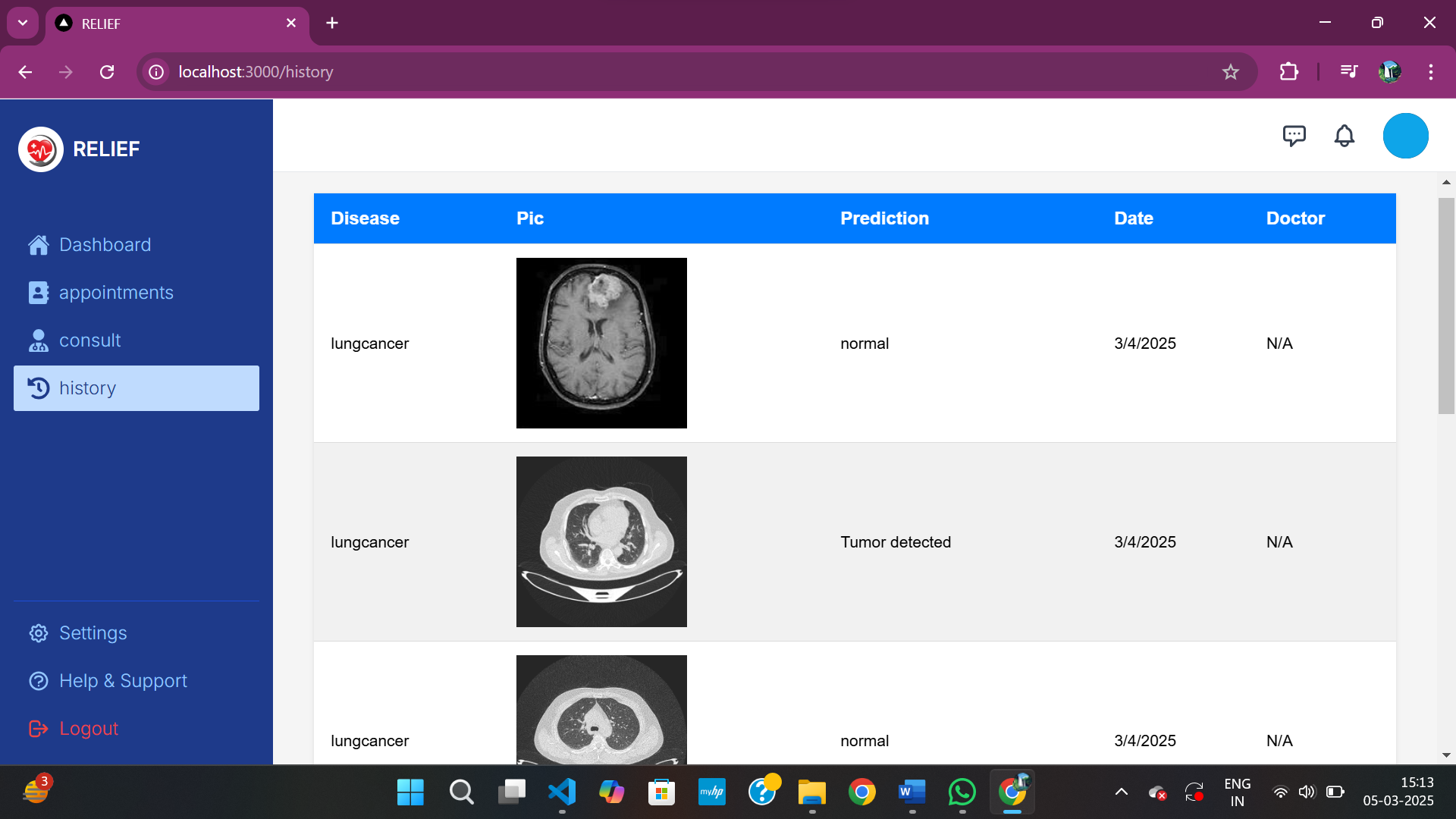
**Fig 21,** Result of Lung MRI-1



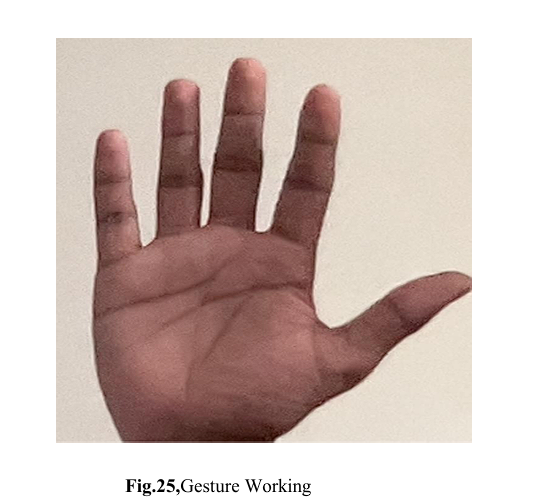
**Fig 22,** Uploading Lung MRI with Tumor



**Fig 23,** Result of Lung MRI-2



**Fig 24,** History Page



Complete algorithm work simple array of size 5 where there are only 1’s or 0’s.

[1,1,1,1,1]

1 indicates that the finger is up.

0 indicates that the finger is down.

in this

1st position of value indicate above thumb

2nd position value indicate as index finger.

3rd position value indicated as middle finger.

4th position value indicated as ring finger.

5th position value indicated as little finger.

Based on the hand gesture we give it converts the gesture into array and perform appropriate action on the presentation according to the user.

The hand Tracking module and hand detection works on this mechanism.

We set our accuracy as 80 to 90 % for proper functioning the presentation using gestures.

The algorithm's precision is essential to discern various hand configurations accurately, translating them into discrete actions within the presentation software. It establishes a reliable interface between the user's hand movements and the control of the presentation slides, ensuring that the gestures are correctly identified and mapped to the intended commands.

Achieving an accuracy level between 80% to 90% implies that the system can effectively interpret hand gestures, enabling presenters to navigate through slides, highlight sections, or perform other actions with a high degree of reliability. This reliability is crucial in real-world scenarios, ensuring that the user experiences a seamless and responsive interaction while delivering presentations.

In essence, this algorithmic approach leverages the simplicity of an array-based representation of hand gestures to facilitate accurate and reliable control over presentation software. By interpreting finger positions and converting them into meaningful commands, this system sets a foundation for intuitive and efficient interaction between the user's hand movements and the digital presentation environment. **Speech Recognition in Gesture Recognition Loop:**

Within the gesture recognition loop, there's a conditional block checking for hand gestures and speech recognition. If hand gestures are detected and certain conditions are met, speech recognition is triggered by calling recognize\_speech(). Depending on the recognized command, various actions are performed such as navigating between slides, deleting annotations, etc.

**Speech Recognition Function (recognize\_speech):**

The function starts by activating the microphone as the audio source using sr.Microphone(). It then adjusts for ambient noise using recognizer.adjust\_for\_ambient\_noise(source, duration=1) to improve recognition accuracy. The listen() method of the recognizer object captures audio input from the microphone. The captured audio is then passed to Google's Speech Recognition API using recognize\_google(audio). If speech is recognized successfully, the recognized text is returned. Otherwise, appropriate error messages are displayed

### LEARNING OUTCOMES

**Integration of Computer Vision and GUI Development:**

The code integrates OpenCV for computer vision tasks like hand tracking and gesture recognition with tkinter for GUI development, allowing for interactive applications.

**Real-time Interaction with Webcam:**

The code captures live video frames from the webcam using OpenCV (cv2.VideoCapture), enabling real-time interaction with the user's hand gestures.

**Hand Gesture Recognition:**

It implements hand gesture recognition using the HandDetector class from the cvzone module, allowing users to control actions based on hand gestures detected in the webcam feed.

**Speech Recognition Integration:**

The code integrates the speech\_recognition library to recognize spoken commands captured from the microphone, providing an alternative mode of interaction alongside hand gestures.

**Dynamic File Manipulation:**

It dynamically renames PNG files in a selected folder based on numerical order, demonstrating file manipulation operations using Python's os module.

**Interactive Presentation Control:**

By combining hand gestures and speech recognition, the code enables users to control a presentation (e.g., navigating between slides, adding annotations) in an interactive and hands-free manner.

1. **CONCLUSION WITH CHALLENGES**

Implementing an interactive slideshow presentation system involves several challenges, particularly in integrating various technologies like hand gesture recognition and speech recognition seamlessly. One challenge lies in ensuring the reliability and accuracy of the gesture and speech recognition mechanisms, as they can be influenced by factors such as background noise, lighting conditions, and accent variations. Overcoming these challenges requires robust error handling mechanisms and continuous improvement in the recognition algorithms to enhance user experience and system usability.

In conclusion, while the presented system demonstrates the potential of combining hand gesture and speech recognition for interactive presentations, there are opportunities for refinement and enhancement. Addressing challenges related to recognition accuracy, system responsiveness, and user interface intuitiveness will be critical for developing a more robust and user-friendly presentation system. With further refinement and iteration, the system can offer an engaging and intuitive user experience, empowering presenters to interact with their slides more seamlessly

––-

## REFERENCES

1. Devivara prasad G, Mr. Srinivasulu M. "Hand Gesture Presentation by Using Machine Learning." September 2022|IJIRT | Volume 9 Issue 4 | ISSN: 2349-6002. https://ijirt.org/master/publishedpaper/IJIRT156612\_PAPER.pdf

2. G.Reethika, P.Anuhya, M.Bhargavi. "SLIDE PRESENTATION BY HAND GESTURE RECOGNITION USING MACHINE LEARNING".(IRJET) e-ISSN: 2395-0056 Volume: 10 Issue: 01 | Jan 2023

https:/ /www .irjet.net/archives/V10/i1/IRJET-V10I1100.pdf

3. Hajeera Khanum, Dr. Pramod H B. "Smart Presentation Control by Hand Gestures Using Computer Vision and Google’s Mediapipe" y (IRJET) e-ISSN: 2395-0056 Volume: 09 Issue: 07 | July 2022.

https:/ /www .irjet.net/archives/V9/i7/IRJET-V9I7482.pdf

4. Salonee Powar, Shweta Kadam, Sonali Malage, Priyanka Shingane. "Automated Digital Presentation Control using Hand Gesture Technique."ITM Web of Conferences 44, 03031 (2022).

<https://www.itm-conferences.org/articles/itmconf/pdf/2022/04/itmconf_icacc2022_03031.pdf>

5. Bobo Zeng, Guijin Wang, Xinggang Lin . "A Hand Gesture Based Interactive Presentation System Utilizing Heterogeneous Cameras." ISSNll1007-0214ll15/18llpp329-336 Volume 17, Number 3, June 2012. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6216765>

[6] Meera Paulson, Nathasha P R, Silpa Davis, Soumya Varma."Smart Presentation Using Gesture Recognition". 2017 IJRTI | Volume 2, Issue 3 | ISSN: 2456-3315 [https://www.ijrti.org/papers/IJRTI1703013.pdf 45](https://www.ijrti.org/papers/IJRTI1703013.pdf%2045)

[7] Rina Damdoo, Kanak Kalyani ,Jignyasa Sanghavi."Adaptive Hand Gesture Recognition System Using Machine Learning Approach".Biosc.Biotech.Res.Comm. Special Issue Vol 13 No 14 (2020) Pp-106-110. <https://bbrc.in/wp-content/uploads/2021/01/13_14-SPL-Galley-proof-026.pdf>

[8] Bhor Rutika, Chaskar Shweta, Date Shraddha, Prof. Auti M. A.4. Power Point Presentation Control Using Hand Gestures Recognition .International Journal of Research Publication and Reviews, Vol 4, no 5, pp 5865-5869, May 2023

<https://ijrpr.com/uploads/V4ISSUE5/IJRPR13592.pdf>

[9] THIN THIN HTOO, OMMAR WIN. Hand Gesture Recognition System for Power Point Presentation.ISSN 2319-8885 Vol.07,Issue.02, February-2018 <https://ijsetr.com/uploads/251346IJSETR16450-43.pdf>

[10] Ram Rajesh J,Sudharshan R,Nagarjunan D,Aarthi R. Remotely controlled PowerPoint presentation navigation using hand gestures