



Mini Project Report On

VoiceVIVA

*Submitted in partial fulfillment of the requirements for the
award of the degree of*

Bachelor of Technology

in

Computer Science & Engineering

By

Neha Davis (U2103153)

Neha Mariam Mathew (U2103154)

Priya Anto (U2103167)

Shreya Sunil (U2103197)

Under the guidance of

Ms. Meenu Mathew

Department of Computer Science & Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Affiliated to APJ Abdul Kalam Technological University)

Rajagiri Valley, Kakkanad, Kochi, 682039

May 2024

CERTIFICATE



RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

*This is to certify that the mini project report entitled "**VoiceVIVA**" is a bonafide record of the work done by **Neha Davis (U2103153)**, **Neha Mariam Mathew (U2103154)**, **Priya Anto (U2103167)**, **Shreya Sunil (U2103197)**, submitted to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2023-2024.*

Project Guide

Ms. Meenu Mathew
Assistant Professor
Dept. of CSE
RSET

Project Coordinator

Dr. Uma Narayanan
Assistant Professor
Dept. of CSE
RSET

HOD

Dr. Preetha K G
Head of the Department
Dept. of CSE
RSET

ACKNOWLEDGEMENTS

We wish to express our sincere gratitude towards Dr P. S. Sreejith, Principal of RSET, and Dr. Preetha K.G., Head of the Department of Computer Science and Engineering for providing us with the opportunity to undertake our mini project, "VoiceVIVA".

We are highly indebted to our project coordinator, **Dr.Uma Narayanan**, Asst Professor, Department of Computer Science and Engineering for her valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Ms. Meenu Mathew** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, We would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Neha Davis

Neha Mariam Mathew

Priya Anto

Shreya Sunil

Abstract

VoiceVIVA is an innovative platform transforming traditional VIVA (oral examination) assessments by allowing users to respond orally, with answers automatically converted to text for verification against an answer key. This process utilizes advanced speech recognition and natural language processing (NLP) techniques to ensure accurate assessment by identifying essential keywords and synonyms, thereby eliminating the need for manual grading and reducing human error.

The platform incorporates gamification elements to enhance user engagement and motivation. Features such as varying difficulty levels, leaderboards, badges, and achievements make the learning process more interactive and enjoyable. By integrating these elements, VoiceVIVA encourages continuous learning and improvement, helping users stay motivated and invested in their studies.

VoiceVIVA's seamless integration of audio input, text conversion, automated verification, and gamification represents a significant advancement in educational technology. This modern approach to VIVA examinations not only improves efficiency and accuracy in grading but also fosters better comprehension and knowledge retention, providing a comprehensive solution for modernizing oral assessments in educational settings.

Contents

Acknowledgements	i
Abstract	ii
List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 Background	1
1.2 Problem Definition	1
1.3 Scope and Motivation	2
1.4 Objectives	2
1.5 Challenges	3
1.6 Assumptions	3
1.7 Societal / Industrial Relevance	4
1.8 Organization of the Report	4
2 Software Requirements Specification	5
2.1 Introduction	5
2.1.1 Purpose	5
2.1.2 Product Scope	5
2.2 Overall Description	6
2.2.1 Product Perspective	6
2.2.2 Product Functions	6
2.2.3 Operating Environment	7
2.2.4 Design and Implementation Constraints	7
2.2.5 Assumptions and Dependencies	8
2.3 External Interface Requirements	9

2.3.1	User Interfaces	9
2.3.2	Hardware Interfaces	10
2.3.3	Software Interfaces	10
2.3.4	Communications Interfaces	11
2.4	System Features	12
2.4.1	Voice to Text and Text to Voice Conversion	12
2.4.2	Question and Answer Generation	13
2.4.3	Keyword and Synonym Checker	14
2.4.4	Score and Feedback Generation	16
2.4.5	Login and Security	17
2.4.6	Difficulty Selection	18
2.4.7	Gaming Features	19
2.4.8	Personalized Report	20
2.5	Other Nonfunctional Requirements	21
2.5.1	Performance Requirements	21
2.5.2	Safety Requirements	22
2.5.3	Security Requirements	22
2.5.4	Software Quality Attributes	23
3	System Architecture and Design	24
3.1	System Overview	24
3.2	Architectural Design	26
3.3	Proposed Methodology/Algorithms	26
3.4	User Interface Design	29
3.5	Database Design	44
3.6	Description of Implementation Strategies	46
3.7	Module Division	52
3.8	Work Schedule - Gantt Chart	56
4	Result and Discussions	58
4.1	Overview	58
4.2	Testing	59
4.3	Discussion	64

5 Conclusion	65
5.1 Conclusion	65
5.2 Future Scope	65
Appendix A: Presentation	67
Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes	96
Vision, Mission, POs, PSOs and COs	2
Appendix C: CO-PO-PSO Mapping	6

List of Figures

3.1	System Architecture Diagram	24
3.2	ER Diagram	26
3.3	Home Page	29
3.4	Login Page	29
3.5	Sign Up Page	30
3.6	To upload image for face recognition after registering the user	30
3.7	Once registration is successful, access to two modes-practice and test	31
3.8	Verifying identity of user	31
3.9	Invalid user authentication	32
3.10	Error message on invalid user authentication	32
3.11	Valid user authentication	33
3.12	On successful user verification-start test	33
3.13	Question generation in Test	34
3.14	Question Generation in test- when the difficulty level is increased	34
3.15	Active Tab detection-warning message	35
3.16	Active Tab Detection-test ends	35
3.17	Timer elapsed in test	36
3.18	Voice to text in test	36
3.19	End page of test with motivational message, personalized feedback and review questions	37
3.20	Review questions in test	37
3.21	A part of end page of test having review question and buttons for try again and leaderboard	38
3.22	Leaderboard	38
3.23	Start practice mode	39
3.24	Question Generation in practice	39
3.25	Voice to Text in practice	40

3.26	Correct Answer Generation, User Answer Evaluation, Feedback and score generation in practice for correct answer	40
3.27	Correct Answer Generation, User Answer Evaluation, Feedback and score generation in practice for wrong answer	41
3.28	End page of practice mode	41
3.29	Detailed personalized report in practice based on your performance in practice, highlighting your strengths and weaknesses	42
3.30	Review Questions in practice, allowing to revise the answers to the questions in one glance	42
3.31	Personal profile of an account showing details of your previous attempts . .	43
3.32	Personal profile of an account showing the performance of the user in a graph format	43
3.33	People	44
3.34	User	44
3.35	Voice to Text Code Snippet	46
3.36	Text to Voice Code Snippet	46
3.37	Question Generation(Practice) Code Snippet	47
3.38	Answer Generation Code Snippet	47
3.39	Score Generation (Practice) Code Snippet	48
3.40	Answer Evaluation (Practice) Code Snippet	48
3.41	Question Generation (Test) Code Snippet	49
3.42	Answer Evaluation (Test) Code Snippet	50
3.43	Flask Integration Code Snippet	51
3.44	Front end Code Snippet	51
3.45	Firebase Code Snippet	51
3.46	Leaderboard Code Snippet	52
3.47	Work Schedule - Gantt Chart	56
4.1	Question generation	59
4.2	Answer generation	59
4.3	End page of test with motivational message, personalized feedback and review questions	60

4.4	Review questions in test	60
4.5	A part of end page of test having review question and buttons for try again and leaderboard	61
4.6	Leaderboard	61
4.7	Detailed personalized report in practice based on your performance in practice, highlighting your strengths and weaknesses	62
4.8	Review Questions in practice, allowing to revise the answers to the questions in one glance	62
4.9	Personal profile of an account showing details of your previous attempts . .	63
4.10	Personal profile of an account showing the performance of the user in a graph format	63

List of Tables

3.1 Module Division	55
-------------------------------	----

Chapter 1

Introduction

1.1 Background

Traditional oral examinations, often referred to as VIVA assessments, have long been a staple in educational evaluation, aimed at assessing students' comprehension, communication skills, and critical thinking abilities. However, these assessments pose various challenges, including time-consuming manual grading processes, subjective evaluation criteria leading to potential bias, and limited scalability and flexibility.

In response to these challenges, VoiceVIVA emerges as an innovative solution leveraging cutting-edge technology to modernize the VIVA examination process. By harnessing advancements in speech recognition, natural language processing (NLP), and automated verification, VoiceVIVA automates the conversion and evaluation of spoken responses, alleviating the burden of manual grading and enhancing objectivity.

Additionally, VoiceVIVA integrates gamification elements such as difficulty selection, leaderboards and badges to foster user engagement and motivation. This aligns with contemporary educational trends emphasizing interactive and immersive learning experiences.

VoiceVIVA represents a significant advancement in assessment methodology, offering a dynamic and efficient alternative to traditional VIVA assessments. By addressing the limitations of manual grading, enhancing objectivity, and promoting user engagement, VoiceVIVA has the potential to revolutionize educational practices and contribute to more effective learning outcomes across diverse educational settings.

1.2 Problem Definition

The aim of the VoiceVIVA project is to address the inefficiencies and subjectivity inherent in traditional oral assessments by automating the process. This solution aims to enhance

accuracy, reduce the burden on educators, and empower students to improve their confidence and performance through automated oral assessment and engaging gamification elements.

1.3 Scope and Motivation

VoiceVIVA encompasses a wide range of functionalities tailored to modernize the oral examination process, primarily within educational contexts. Its core features include capturing spoken responses, converting them into text, and automating the verification process through speech recognition and natural language processing. The system's scope extends to various educational disciplines and levels, accommodating different subjects and difficulty levels to cater to diverse learning needs. Additionally, VoiceVIVA integrates gamification elements to enhance user engagement and motivation, further enriching the assessment experience. Overall, the scope of VoiceVIVA is comprehensive, aiming to revolutionize assessment methodologies and improve learning outcomes across educational settings.

The motivation behind VoiceVIVA stems from the need to address the limitations and challenges associated with traditional oral examinations. By automating and streamlining the assessment process, VoiceVIVA reduces the burden of manual grading, minimizes human error, and ensures consistency and objectivity in evaluation. Furthermore, the integration of gamification elements serves to enhance user motivation and participation, transforming the assessment experience into an engaging and interactive journey. Ultimately, VoiceVIVA's motivation lies in its potential to facilitate more effective learning outcomes by providing a dynamic and innovative approach to conducting oral assessments.

1.4 Objectives

- Facilitate VIVA Practice: Provide students with a platform where they can simulate VIVA assessments, allowing them to practice responding to questions in a controlled environment and familiarize themselves with the format and expectations of oral examinations.
- Build Confidence: Design features within the system that offer constructive feedback

and encouragement to students, helping to boost their confidence and alleviate anxiety associated with oral assessments.

- Offer Personalized Learning: Utilize adaptive learning algorithms to tailor practice sessions to the individual needs and skill levels of each student, providing targeted support and guidance to enhance their preparation for VIVA examinations.
- Cultivate Communication Skills: Incorporate interactive exercises and prompts that encourage students to articulate their thoughts and ideas effectively, helping them to develop and refine their communication skills essential for successful VIVA assessments.
- Enhance Assessment Efficiency: Streamline the process of conducting VIVA assessments by automating the transcription and verification of spoken responses, reducing the time and effort required for manual grading.
- Foster Active Learning: Engage users in the learning process by providing immediate feedback on their responses, encouraging reflection and reinforcement of concepts covered during the VIVA examination.

1.5 Challenges

Challenges in developing VoiceVIVA include fine-tuning the speech recognition algorithms to accurately capture diverse accents and speech patterns. Additionally, ensuring robust keyword and synonym checking algorithms poses a challenge, requiring extensive testing and refinement to achieve accurate evaluations while minimizing false positives and negatives. Integrating gamification elements seamlessly into the assessment process while maintaining its educational integrity is also a significant hurdle.

1.6 Assumptions

- The accuracy of speech-to-text conversion algorithms is reliable enough to ensure the fidelity of users' oral responses when converted into written text.
- Users have access to a reliable internet connection and compatible devices (such as computers or smartphones) to access the VoiceVIVA platform.

- The automated verification process effectively identifies both exact matches and semantic similarities between user responses and the answer key, ensuring fair and comprehensive evaluations.

1.7 Societal / Industrial Relevance

VoiceVIVA significantly benefits students by revolutionizing assessment in both educational and professional contexts. In education, it provides immediate feedback, fostering active learning, and reducing educator workload. In professions, it streamlines certification processes, ensuring fair evaluations and validating students' skills. Its emphasis on inclusivity ensures equitable access to educational opportunities for all students, regardless of background or learning needs. Overall, VoiceVIVA's impact lies in empowering students, enhancing their learning experiences, and preparing them for success.

1.8 Organization of the Report

The report begins with an Introduction section that delineates the Problem Definition, identifying the precise scope and motivation behind the project, setting the stage for the subsequent discussions. Here, the Objectives and Challenges are outlined, providing a clear understanding of the project's goals and potential hurdles. Following this, the report delves into Societal and Industrial Relevance, highlighting the broader impact and significance of the proposed solution. Moving forward, the System Architecture and Design section offers a comprehensive overview of the project's technical framework, including a System Overview, Architectural Design, Proposed Methodology/Algorithms, User Interface Design, and Database Design, each elucidating critical aspects of the system's structure and functionality. Within this section, Description of Implementation Strategies elucidates how the proposed solution will be realized, while Module Division delineates the breakdown of the project into manageable components. Concluding the report, the Work Schedule - Gantt Chart presents a visual representation of the project timeline, ensuring transparency and accountability in project management. This structured road map guides stakeholders through each phase of the project, ensuring clarity of purpose, technical feasibility, and adherence to deadlines.

Chapter 2

Software Requirements Specification

2.1 Introduction

2.1.1 Purpose

This Software Requirements Specification (SRS) document pertains to VoiceVIVA version 1.0, an innovative interactive website aimed at revolutionizing the process of conducting VIVA (oral examination) assessments. The scope of this SRS encompasses the entire VoiceVIVA system, detailing its core functionalities, features, and requirements necessary for its successful implementation. This document provides a comprehensive outline of the software requirements for VoiceVIVA, including its automated oral assessment capabilities, speech recognition algorithms, automated verification processes, and gamification elements aimed at enhancing user engagement and motivation.

2.1.2 Product Scope

VoiceVIVA represents a paradigm shift in the realm of educational assessment, offering a sophisticated and efficient solution for conducting oral examinations. Tailored specifically for educational environments where oral assessments are commonplace, VoiceVIVA streamlines the assessment process by enabling users to respond verbally to questions, with their answers seamlessly converted into text format for comparison against predefined answer keys. Leveraging state-of-the-art speech recognition algorithms and advanced natural language processing techniques, VoiceVIVA ensures high accuracy and efficiency in assessment procedures. The primary objectives of VoiceVIVA encompass various facets aimed at enhancing the assessment methodology and improving learning outcomes. Firstly, it aims to automate and optimize assessment procedures, reducing manual grading efforts and ensuring consistency and objectivity in evaluation. Additionally, VoiceVIVA incorporates

gamification elements such as difficulty settings, leaderboards, badges, and achievements, fostering user engagement and motivation. These features not only incentivize users to actively participate in assessments but also provide a dynamic and interactive learning environment conducive to knowledge retention and comprehension.

2.2 Overall Description

2.2.1 Product Perspective

VoiceVIVA addresses the pressing need for innovative approaches to educational assessment, especially in environments where manual grading of oral exams persists. Leveraging cutting-edge speech recognition and natural language processing technologies, VoiceVIVA streamlines the assessment process by converting oral responses into text for automated evaluation, significantly reducing grading time and promoting objectivity. Its development underscores a commitment to revolutionizing traditional assessment methods and leveraging technology to enhance educational practices.

While currently a self-contained product, VoiceVIVA's future potential lies in its seamless integration into existing educational systems through integration APIs, thereby improving accessibility and scalability. This integration positions VoiceVIVA as a pivotal component within broader educational ecosystems, complementing existing platforms and driving the modernization of assessment practices. By adapting to the evolving needs of educators and learners, VoiceVIVA aspires to play a vital role in shaping the future of educational assessment, fostering enhanced learning outcomes and facilitating student success.

2.2.2 Product Functions

1. Oral Response Capture and Speech-to-Text Conversion: Enable users to provide oral responses to questions through the platform. Implement speech recognition algorithms to convert oral responses into written text for further processing.
2. Text-to-Speech for Questions: Provide a feature for converting text-based questions into spoken form for user interaction.
3. Question and Answer Generation: Develop functionality for generating a variety of

questions and model answers to diversify assessments.

4. Automated Evaluation and Feedback: Include mechanisms for automated scoring based on response accuracy and predefined criteria. Implement feedback generation to provide users with constructive feedback on their performance.
5. Engagement Features and Scalability: Integrate difficulty selection options, personalized reports, leaderboards, badges, and user engagement metrics to enhance user experience. Ensure scalability to accommodate a growing user base and increasing data load, while also maintaining cross-platform compatibility and implementing necessary security measures.

2.2.3 Operating Environment

VoiceVIVA operates on standard computing hardware typically found in educational institutions, offices, and personal computers. The minimum hardware requirements include a desktop or laptop computer with a modern processor and at least 4GB of RAM. For optimal performance, it is recommended to have a faster processor and higher RAM capacity. The software is compatible with various operating systems. It operates natively on these systems to ensure smooth operation and compatibility.

VoiceVIVA may require access to certain software components and applications for enhanced functionality and compatibility. This includes web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari (compatible with the latest stable versions). It also utilizes speech recognition libraries and APIs for converting spoken responses into text format. Database systems may be needed for storing application-related data. Stable internet connectivity is essential for accessing the application, submitting responses, and receiving feedback. By ensuring compliance with these operating environment requirements, users can optimize performance and ensure compatibility across different platforms and software components.

2.2.4 Design and Implementation Constraints

VoiceVIVA's development process is governed by several constraints that dictate the available options for developers. These constraints encompass corporate policies, regulatory requirements, technological considerations, and design conventions. Compliance

with corporate policies and regulatory standards is paramount to ensure data privacy and security. Developers must also account for hardware limitations, including timing requirements and memory constraints, to ensure optimal performance across various computing platforms.

Integration with external applications or systems necessitates compatibility with existing software interfaces and protocols. Specific technologies, tools, and databases mandated by the client or organizational standards must be utilized. Security considerations, such as encryption protocols and access control mechanisms, are crucial to mitigate risks effectively. Adherence to established design conventions and programming standards ensures consistency and maintainability.

Communication protocols compatible with external interfaces and language localization requirements may also impact design and implementation decisions. By addressing these constraints, developers can navigate limitations and deliver a robust and compliant solution that aligns with stakeholders' needs and expectations.

2.2.5 Assumptions and Dependencies

Assumed Factors

1. Accuracy and Performance of Speech Recognition Algorithms: The effectiveness of VoiceVIVA heavily relies on the accuracy and performance of the speech recognition algorithms used to convert spoken responses into written text. Any limitations or inaccuracies in these algorithms could impact the overall functionality and reliability of the system.
2. Availability and Reliability of Synonym Checker Component: Since VoiceVIVA utilizes sophisticated algorithms to analyze for the presence of similar words, including synonyms, the availability and reliability of the synonym checker component are crucial. Assumptions about its effectiveness and accessibility could affect the accuracy of evaluations.
3. Scalability of the System: As the user base grows, scalability becomes a crucial factor. Assumptions about the system's ability to handle increased loads of users and assessments need to be considered. Any limitations in scalability could hinder the system's performance and user experience.

4. Compatibility with Different Accents and Languages: VoiceVIVA's ability to accurately transcribe spoken responses may vary depending on the accent and language of the users. Assumptions about the system's compatibility with different accents and languages could impact its usability and effectiveness in diverse educational settings.
5. User Engagement with Gamification Elements: VoiceVIVA incorporates gamification elements to enhance user engagement and motivation. Assumptions about users' response to these gamification features and their effectiveness in improving learning outcomes may influence the overall success of the system.

Dependencies

1. Gemini API for Question and Answer Generation: VoiceVIVA relies on the Gemini API for generating questions and answers. Any changes, limitations, or unavailability of this API could directly impact the functionality and content of VoiceVIVA. Dependencies on third-party APIs introduce risks related to their maintenance, updates, and availability.
2. External Speech Recognition Components: While not explicitly stated, VoiceVIVA depends on speech recognition algorithms. These algorithms may be sourced from external libraries, frameworks, or services. Any changes or limitations in these components could affect the accuracy and performance of the speech-to-text conversion process, impacting the overall functionality of VoiceVIVA.
3. Database for User Authentication: The login and register pages in VoiceVIVA depend on a database for user authentication. Any issues with the database, such as connectivity problems, data corruption, or unauthorized access, could impact the user registration and login process, hindering access to the platform's features.

2.3 External Interface Requirements

2.3.1 User Interfaces

VoiceVIVA's user interfaces are meticulously designed to ensure seamless interaction, adhering to established GUI standards and product style guides. With a clean layout

and intuitive design, users experience efficient navigation and a focused user experience, promoting usability and industry best practices. The platform offers two main interfaces: the oral response interface for articulating answers orally, utilizing speech recognition algorithms, and the feedback interface for delivering assessment results and personalized reports. Accessible via a web-based platform, VoiceVIVA ensures cross-platform compatibility, standardized buttons, and straightforward navigation, contributing to a positive and engaging user experience.

Through adherence to design principles and industry standards, VoiceVIVA provides a cohesive and user-centric interface. With consistent navigation and clear presentation of questions and feedback, the platform aims to enhance usability and engagement, promoting a positive assessment experience for users across various devices and browsers.

2.3.2 Hardware Interfaces

VoiceVIVA's hardware interface serves as the crucial link between the software and various hardware components, facilitating essential functionalities of the system. Operating primarily on standard computing hardware like desktops or laptops, the platform supports a range of input/output devices such as microphones for audio input during oral assessments and speakers or headphones for audio output.

Communication with hardware devices is seamless, with VoiceVIVA capturing audio input from users during assessments and converting it into text format for analysis. The software also manages control aspects such as audio settings and device configurations. Leveraging standard communication protocols like USB for external devices and OS-specific protocols for internal communication, VoiceVIVA ensures smooth interaction with hardware components, enhancing operational efficiency and user experience.

2.3.3 Software Interfaces

VoiceVIVA relies on seamless integration with the Gemini database management system, playing a pivotal role in the overall software architecture. The Gemini database serves as a robust and efficient repository for storing, managing, and retrieving essential data integral to VoiceVIVA's functionality. User information, assessment details, and gamification metrics are stored and organized within Gemini, allowing for real-time access

and updates. This integration ensures the secure and streamlined handling of diverse data sets, contributing to the reliability and scalability of VoiceVIVA.

The interaction between VoiceVIVA and Gemini involves the website capturing spoken responses from users, which are then processed and converted into text format using advanced speech recognition algorithms and NLP techniques. The converted text, along with relevant user data and assessment information, is seamlessly integrated with Gemini. The database facilitates the storage and retrieval of this data, enabling the automated verification process by comparing the converted text against a generated answer key. The incorporation of Gemini plays a crucial role in eliminating the need for manual grading, reducing human error, and enhancing objectivity in the assessment process.

Additionally, Gemini supports the gamification elements embedded in VoiceVIVA, ensuring efficient tracking and updating of user achievements, badges, leaderboards, and difficulty levels. This integration enhances user engagement and motivation, providing a dynamic and personalized experience for individuals progressing through assessment levels. The reliance on Gemini as the database component establishes a cohesive and efficient framework for VoiceVIVA's functionalities. It aligns with software requirements specifications by ensuring data integrity, security, and accessibility. The use of Gemini as a central component underscores the commitment to a robust and scalable infrastructure, contributing to the overall success of VoiceVIVA in modernizing the VIVA examination process.

2.3.4 Communications Interfaces

VoiceVIVA employs communication interfaces to facilitate its functionalities smoothly. The Speech Input Interface captures user responses in real-time, supporting raw audio data and secure channels to maintain data integrity. Once captured, the Text Output Interface converts spoken responses into text, utilizing standard protocols like HTTP or WebSocket while encrypting data for confidentiality.

Integrating with the Gemini API for question and answer generation, VoiceVIVA uses an interface that communicates in JSON or XML formats over secure HTTP or HTTPS protocols. Authentication credentials are securely managed to access the Gemini API endpoints, ensuring data protection during interactions.

VoiceVIVA may also interact with external synonym checker services through an in-

tegration interface, ensuring compatibility and maintaining security with secure communication channels and data encryption. For user authentication, VoiceVIVA employs a dedicated interface to handle login and registration securely. Utilizing HTTPS, it verifies user credentials against the database and

2.4 System Features

2.4.1 Voice to Text and Text to Voice Conversion

2.4.1.1 Description and Priority

VoiceVIVA prioritizes Voice to Text and Text to Voice Conversion, essential for seamless oral assessments. Its high priority stems from its core functionality, enabling effective user engagement.

- Benefit: 8 - Implementing this feature improves user experience and accessibility.
- Penalty: 3 - Inadequate implementation may cause minor usability issues.
- Cost: 6 - Moderate development and maintenance expenses are involved.
- Risk: 7 - Not meeting user expectations poses a risk of dissatisfaction and reduced adoption.

2.4.1.2. Stimulus/Response Sequences

1. User speaks into the microphone to provide an oral response during an assessment.
2. The system captures the spoken response and converts it into text format.
3. The converted text is displayed on the screen for the user to review.
4. Alternatively, the user selects a text-based response, triggering the system to convert it into synthesized speech (Text to Voice).
5. The system audibly presents the converted text-based response to the user.
6. Users may adjust volume settings or pause/play the synthesized speech as needed.

2.4.1.3. Functional Requirements

- REQ-1: Voice Input Processing

VoiceVIVA must accurately capture and process spoken responses using advanced speech recognition algorithms, robust against background noise.

- REQ-2: Text Output Synthesis

VoiceVIVA synthesizes text-based responses into clear and natural-sounding speech.

- REQ-3: Validation of Inputs

VoiceVIVA validates both voice and text inputs for completeness, correctness, and relevance, providing appropriate feedback for invalid inputs.

- REQ-4: Compatibility and Integration

VoiceVIVA is compatible with various devices and operating systems, seamlessly integrating with external speech recognition and text-to-speech engines for enhanced performance across diverse user environments.

2.4.2 Question and Answer Generation

2.4.2.1. Description and Priority

VoiceVIVA prioritizes the innovative generation of questions and answers to ensure a dynamic and comprehensive assessment experience. The system employs cutting-edge speech recognition and natural language processing (NLP) techniques to capture spoken responses and convert them into text.

Priority Component Ratings:

- Benefit: 8- Enhances assessment efficiency with personalized experiences.
- Penalty: 3- Challenges may arise in aligning questions with diverse responses
- Cost: 6- Involves development and refinement of question generation algorithms.
- Risk: 7- Requires rigorous testing to ensure accurate alignment of questions and answers

2.4.2.2. Stimulus/Response Sequences

1. Users respond orally, captured by advanced speech recognition algorithms.
2. Speech is converted to text using cutting-edge technology.
3. Converted text is compared to an answer key automatically.
4. Automated verification checks for keywords and similar words, ensuring accuracy.
5. Gamification elements like leaderboards, badges, and achievements engage users.
6. Gamification boosts motivation, encouraging performance improvement.

2.4.2.3. Functional Requirements

- REQ-1: Speech Recognition and Conversion

VoiceVIVA must capture spoken responses from users.

- REQ-2: Text Conversion

The system should employ advanced speech recognition algorithms to convert spoken responses into written text format.

- REQ-3: Automated Verification

VoiceVIVA must compare converted text against a generated answer key.

- REQ-4: Keyword Checking

The automated verification process should incorporate keyword checking.

- REQ-5: Similar Word Analysis

Sophisticated algorithms should analyze for the presence of similar words, including synonyms.

2.4.3 Keyword and Synonym Checker

2.4.3.1. Description and Priority

The Keyword and Synonym Checker in VoiceVIVA ensures accurate evaluation by identifying keywords and synonyms, crucial for assessment reliability. High priority for its direct impact on assessment accuracy.

Priority Component Ratings:

- Benefit: 9 - Enhances assessment accuracy and reliability.
- Penalty: 8 - Inaccurate evaluation could lead to misjudgment of user responses.
- Cost: 6 - Development and integration may require moderate resources.
- Risk: 7 - There is a risk of misidentifying keywords or synonyms, affecting assessment outcomes.

2.4.3.2. Stimulus/Response Sequences

1. User speaks response.
2. VoiceVIVA captures and converts to text.
3. VoiceVIVA generates the correct answer.
4. Keyword and Synonym Checker analyzes both responses.
5. System checks synonyms for keywords.
6. VoiceVIVA provides feedback on response accuracy.

2.4.3.3. Functional Requirements

- REQ-1: Keyword Identification

VoiceVIVA will utilize NLP techniques to identify essential keywords from user responses and generated answers. When keywords cannot be identified due to unclear responses or technical constraints, the system prompts users for clarification to improve accuracy.

- REQ-2: Synonym Detection

VoiceVIVA will incorporate a synonym detection mechanism utilizing a comprehensive database. If synonyms cannot be accurately detected, they will not be considered during evaluation to avoid ambiguity.

- REQ-3: Feedback Generation

VoiceVIVA will provide clear and concise feedback on keyword and synonym presence in user responses. In case of feedback generation failure due to system errors, users will receive an error message prompting them to retry or seek assistance.

2.4.4 Score and Feedback Generation

2.4.4.1. Description and Priority

VoiceVIVA's Score and Feedback Generation feature assesses user performance, offering automated scores and detailed feedback to enhance learning. It's a high-priority feature, providing personalized insights for improvement.

Priority Component Ratings:

- Benefit: 9 - Enhances learning with personalized feedback, promoting improvement.
- Penalty: 3 - Its absence doesn't affect function but enhances assessment.
- Cost: 7 - Moderately resource-intensive but justifies benefits to learning outcomes.
- Risk: 6 - Low risk, builds on existing features, unlikely to pose major challenges.

2.4.4.2. Stimulus/Response Sequences

1. User completes the assessment session.
2. System starts the automated scoring and feedback generation process.
3. User awaits the assessment results.
4. System calculates the score and provides detailed feedback.
5. User reviews the assessment results and feedback.

2.4.4.3. Functional Requirements

- REQ-1: Automated Scoring and Error Handling:

The system must automatically score user responses, logging errors and displaying user-friendly messages in case of issues.

- REQ-2: Feedback Generation and Presentation:

Automated feedback must be generated based on user performance, employing sophisticated algorithms for multifaceted evaluation. The system should ensure real-time presentation of feedback, addressing any delays promptly.

- REQ-3: User-Friendly Language and Accessibility:

The generated feedback should use constructive and clear language. The system must ensure easy accessibility to feedback through a user-friendly interface. Additionally, real-time updates on leaderboards should be provided, with user-friendly messages in case of delays.

2.4.5 Login and Security

2.4.5.1. Description and Priority

The Login and Security feature is of high priority due to its critical role in safeguarding user data and maintaining system integrity. Ensuring secure access to VoiceVIVA not only protects user privacy but also fosters trust and confidence in the platform, outweighing the moderate costs and risks associated with implementation.

Priority Component Ratings:

- Benefit: 9 - Ensures data security and user privacy.
- Penalty: 7 - Risk of data breaches or unauthorized access without security measures.
- Cost: 6 - Moderate costs associated with implementing strong encryption protocols.
- Risk: 8 - High risk of compromising user data and system integrity.

2.4.5.2. Stimulus/Response Sequences

1. User initiates the login process by entering their username and password.
2. The system verifies the provided credentials against the stored user database.
3. If the credentials are valid, the system grants access to the user account and navigates to the user dashboard.

4. In case of invalid credentials, the system displays an error message prompting the user to re-enter their login information.
5. After a successful login, the system may prompt the user to enable additional security measures such as password change.

2.4.5.3. Functional Requirements

- REQ-1: User Authentication

The system provides a secure login interface for users to enter their credentials. It authenticates user information against the stored database and prompts for accurate login details if invalid.

- REQ-2: System Integrity

Successful authentication grants users access to their accounts and directs them to the user dashboard.

- REQ-3: Administrative Tools

Administrators have efficient tools to manage user accounts, permissions, and security settings, ensuring control over system access.

2.4.6 Difficulty Selection

2.4.6.1. Description and Priority

The difficulty selection feature in VoiceVIVA allows users to customize the challenge level of their assessments, tailoring the experience to their individual proficiency and learning goals. This functionality is integral to the software's user-centric approach and aligns with the educational objective of accommodating diverse skill levels.

Priority Component Ratings:

- Benefit: 9 - Enhances user engagement and adaptability, catering to a diverse user base with varying levels of knowledge and confidence.
- Penalty: 2 - Reduced user motivation and engagement.
- Cost: 5 - The development and implementation costs are moderate, considering the need for a dynamic question pool at different difficulty levels.

- Risk: 3 - Ensuring the accuracy of difficulty levels or addressing user preferences.

2.4.6.2. Stimulus/Response Sequences

1. User initiates the assessment and reaches the difficulty selection screen.
2. VoiceVIVA presents difficulty options (easy, intermediate, hard) for the user to choose from.
3. User responds orally to the generated questions within the chosen difficulty level.

2.4.6.3. Functional Requirements

- REQ-1: User-Selectable Difficulty Levels

VoiceVIVA should allow users to choose from multiple difficulty levels when engaging in assessments. These levels may include "easy," "intermediate," and "hard," catering to varying levels of proficiency and ensuring a personalized learning experience.

- REQ-2: Difficulty Level Indicators

Clear indicators should be provided to users, displaying the selected difficulty level throughout the assessment process.

- REQ-3: Gamified Progression

The difficulty selection process should be seamlessly integrated into the gamification elements of VoiceVIVA.

2.4.7 Gaming Features

2.4.7.1. Description and Priority

Enhances user engagement and motivation with elements like difficulty selection, leaderboards, badges, and achievements. Medium priority for boosting user satisfaction and platform effectiveness.

Priority Component Ratings:

- Benefit: 9 - Boosts user engagement, improving learning outcomes.
- Penalty: 6 - Lack of gaming features may reduce user interest.

- Cost: 7 - Moderate resources needed for development.
- Risk: 5 - Poorly implemented features risk user disengagement.

2.4.7.2. Stimulus/Response Sequences

1. Users engage by responding to VoiceVIVA questions during assessments.
2. VoiceVIVA evaluates user responses for accuracy and completeness.
3. User progress is updated based on assessment results, earning points and badges.
4. Leaderboard ranks change dynamically with user performance relative to others.

2.4.7.3. Functional Requirements

- REQ-1: Evaluate user responses for performance based on accuracy, completeness, and timeliness. Provide feedback for errors and update progress with earned points and badges.
- REQ-2: Define badges and achievements to encourage engagement and progress. Award them upon reaching specific milestones or accomplishments. Notify users and allow them to view earned badges and achievements in their profiles.
- REQ-3: Maintain leaderboards to display user performance rankings. Update positions dynamically based on performance and points earned. Enable users to track progress and ensure fairness in calculations, considering factors like assessment difficulty and user activity.

2.4.8 Personalized Report

2.4.8.1. Description and Priority

Personalized Report feature holds medium priority as it offers valuable insights to users for self-assessment and improvement, enhancing the overall learning experience, but it may not be as critical as features directly impacting real-time assessment or system performance.

Priority Component Ratings:

- Benefit: 7 - This feature significantly enhances the learning experience.
- Penalty: 4 - No direct impact on real-time assessment, yet boosts user engagement.
- Cost: 6 - Moderate development and maintenance costs are required.
- Risk: 5 - The risk is mitigated by testing and feedback.

2.4.8.2. Stimulus/Response Sequences

1. User initiates a request for a personalized report on VoiceVIVA.
2. System processes user data, including past responses and assessment history.
3. User accesses and reviews the generated personalized report.
4. System provides comprehensive performance feedback.
5. Users can save or share their reports for future reference or collaboration purposes.

2.4.8.3. Functional Requirements

- REQ-1: Automated Report Generation:

System must generate error-free personalized reports. In case of errors, it should display user-friendly messages.

- REQ-2: Performance Summary and Consistency:

Reports must include accurate performance summaries. System should handle data inconsistencies with clear error messages.

- REQ-3: Historical Performance Tracking:

Users must review historical reports seamlessly. System should validate inputs for effective tracking.

2.5 Other Nonfunctional Requirements

2.5.1 Performance Requirements

Normal Usage Conditions: Achieve an average response time of 1.5 seconds for processing oral responses to ensure prompt user feedback. Peak Load Conditions: Support a

minimum of 150 concurrent users with no more than a 10 percentage increase in average response time to ensure scalability and consistent performance. Real-Time Processing: Attain a response time of less than 1 second for capturing, processing, and converting oral responses into written text, ensuring immediate feedback in real-time scenarios.

2.5.2 Safety Requirements

VoiceVIVA must adhere to safety requirements to mitigate potential risks and ensure user well-being. VoiceVIVA prioritizes safety to mitigate risks and ensure user well-being. The following features address safety concerns associated with the use of the product:

- Data Security: Protects user data through encryption protocols for transmission and storage.
- Accuracy: Ensures accurate assessments via rigorous testing to maintain reliability.
- Accessibility: Adheres to accessibility standards for equitable access to all users.
- Misuse Prevention: Incorporates measures to prevent unethical use, including user authentication and access controls.
- Regulatory Compliance: Ensures adherence to relevant safety regulations and standards.

By addressing these safety requirements, VoiceVIVA aims to mitigate potential risks and ensure a safe and secure user experience. Compliance with external policies, regulations, and safety certifications further reinforces the commitment to user safety and well-being.

2.5.3 Security Requirements

Ensuring the security and privacy of user data is of utmost importance for VoiceVIVA. The following requirements are specified to address security and privacy issues surrounding the use of the product and the protection of data. VoiceVIVA enforces stringent security constraints, including:

- Data Encryption: All user-Platform communications and stored data are encrypted to safeguard sensitive information.

- User Authentication: A robust authentication system mandates secure user identity verification before accessing VoiceVIVA.
- Data Confidentiality: Clear data ownership and confidentiality agreements ensure compliance with privacy laws and protect user data integrity.

By incorporating these security and privacy requirements, VoiceVIVA can offer users a trustworthy and secure platform while maintaining compliance with relevant regulations and industry best practices.

2.5.4 Software Quality Attributes

- Usability: Prioritize a user-friendly interface for a smooth user experience.
- Reliability: Aim for consistent performance, prioritizing reliability.
- Maintainability: Structure the codebase for easy updates, ensuring high maintainability.
- Scalability: Handle increased user demand without performance degradation, emphasizing vital scalability.
- Interoperability: Ensure compatibility across platforms, prioritizing interoperability.
- Performance Efficiency: Optimize responsiveness for user productivity and satisfaction.

Chapter 3

System Architecture and Design

3.1 System Overview

VoiceVIVA is a state-of-the-art interactive website designed to modernize the process of conducting oral examinations (VIVAs) using innovative speech recognition technology, automated verification, and gamification elements. This system aims to enhance user engagement, streamline assessment processes, and provide valuable feedback to users.

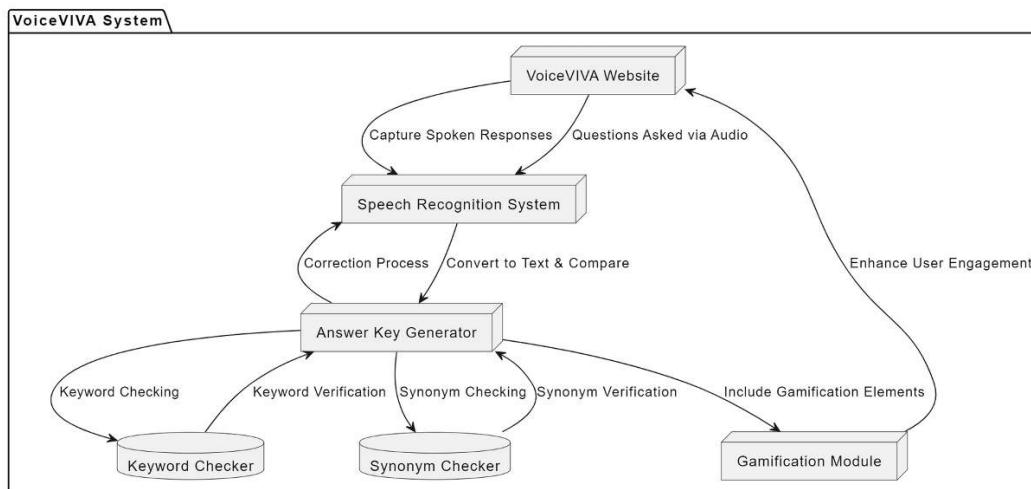


Figure 3.1: System Architecture Diagram

1. User Interaction and Registration:

Users interact with VoiceVIVA through a user-friendly web interface. They register on the platform by providing necessary details such as name, username, email and password.

2. Assessment Setup:

After registration, users can select the subject area and they also have the option to choose the difficulty level of the assessment (easy, medium, hard) based on their proficiency

and learning objectives.

3. Question Presentation:

The assessment begins with VoiceVIVA presenting questions via recorded audio to the user. Questions are designed to be clear and concise, covering relevant topics within the chosen subject area.

4. Spoken Response Capture:

Users respond to the questions orally using a microphone-enabled interface on the website. The system captures these spoken responses using advanced speech recognition technology, converting the spoken words into written text in real-time.

5. Text Conversion and Processing:

The converted text responses are then processed for analysis and evaluation. VoiceVIVA's speech-to-text conversion ensures accurate transcription of user responses, which is essential for subsequent verification against the answer key.

6. Automated Verification Process:

An answer key generator powered by natural language processing (NLP) techniques is employed to evaluate the correctness of user responses.

This verification process includes:

Keyword Checking: Ensuring that essential keywords relevant to the question are present in the response.

Synonym Checker: Analyzing the response for the use of synonymous terms or related words that convey the intended meaning.

Semantic Analysis: Examining the overall semantic coherence and relevance of the response to the question.

7. Assessment Results and Feedback:

Once the verification process is complete, users receive instant feedback on their performance. Feedback includes details on correct answers, areas for improvement, and overall scores. Users can review their responses and understand the reasoning behind the assessment outcomes, facilitating continuous learning.

8. Gamification Elements:

VoiceVIVA incorporates gamification features to enhance user engagement and motivation.

Gamification elements include:

Difficulty Levels: Users can progress through different difficulty levels, earning points and achievements.

Leaderboards: Displaying top performers encourages healthy competition and participation.

Badges and Rewards: Recognizing user achievements and milestones further incentivizes active participation.

3.2 Architectural Design

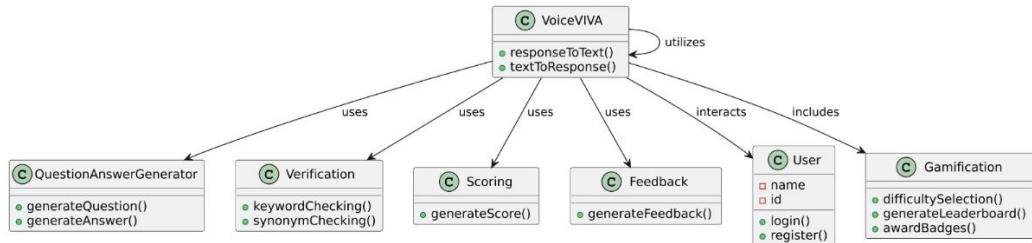


Figure 3.2: ER Diagram

3.3 Proposed Methodology/Algorithms

VoiceVIVA incorporates several methodologies and algorithms to achieve its core functionalities, including speech recognition, text processing, question generation, answer evaluation, and scoring.

1. Speech Recognition:

Methodology: Utilizes a speech recognition library to convert spoken audio into text format.

Algorithm: The algorithm involves using a pre-trained model such as Google's Speech Recognition API to transcribe audio input to text.

Input: Audio data from microphone

Output: Transcribed text

- Capture audio input from the microphone.
- Apply speech recognition algorithm to convert audio to text.
- Return the transcribed text as output.

2. Text-to-Speech Conversion:

Methodology: Converts text into audible speech output.

Algorithm: Uses a text-to-speech (TTS) synthesis engine to generate spoken audio from text.

Input: Text to be converted to speech

Output: Audio output

- Initialize the text-to-speech engine.
- Provide the input text to the engine.
- Generate and play the audio output.

3. Question Generation:

Methodology: Dynamically creates assessment questions based on subject and difficulty level.

Algorithm: Involves selecting or generating questions from predefined datasets or algorithms.

Input: Subject, Difficulty Level

Output: Generated Question

- Retrieve or generate question templates based on the subject.

- Customize the question based on the difficulty level (e.g., easy, medium, hard).
- Return the generated question as output.

4. Answer Evaluation:

Methodology: Compares user-provided answers against correct answers to assess correctness.

Algorithm: Implements keyword matching, synonym checking, or semantic analysis to evaluate answers.

Input: User Response, Correct Answer

Output: Boolean (Correct or Incorrect)

- Preprocess user response and correct answer.
- Apply evaluation criteria (e.g., keyword matching, synonym analysis).
- Determine if the user response matches the correct answer.
- Return True if correct, False otherwise.

5. Score Generation:

Methodology: Computes scores based on evaluation results and difficulty level.

Algorithm: Assigns scores to user responses based on correctness and difficulty.

Input: Evaluation Result (Correct/Incorrect), Difficulty Level

Output: Score

- Determine base score based on correctness.
- Adjust score based on difficulty level.
- Return the final score as output.

3.4 User Interface Design

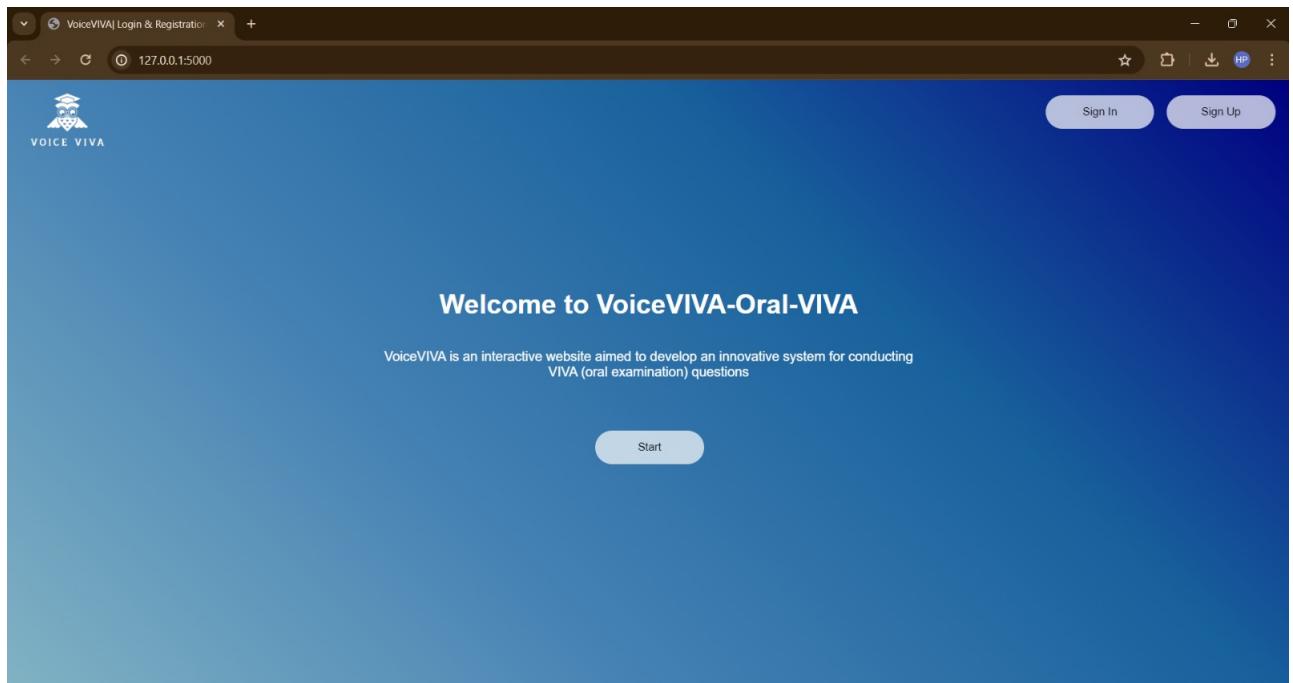


Figure 3.3: Home Page

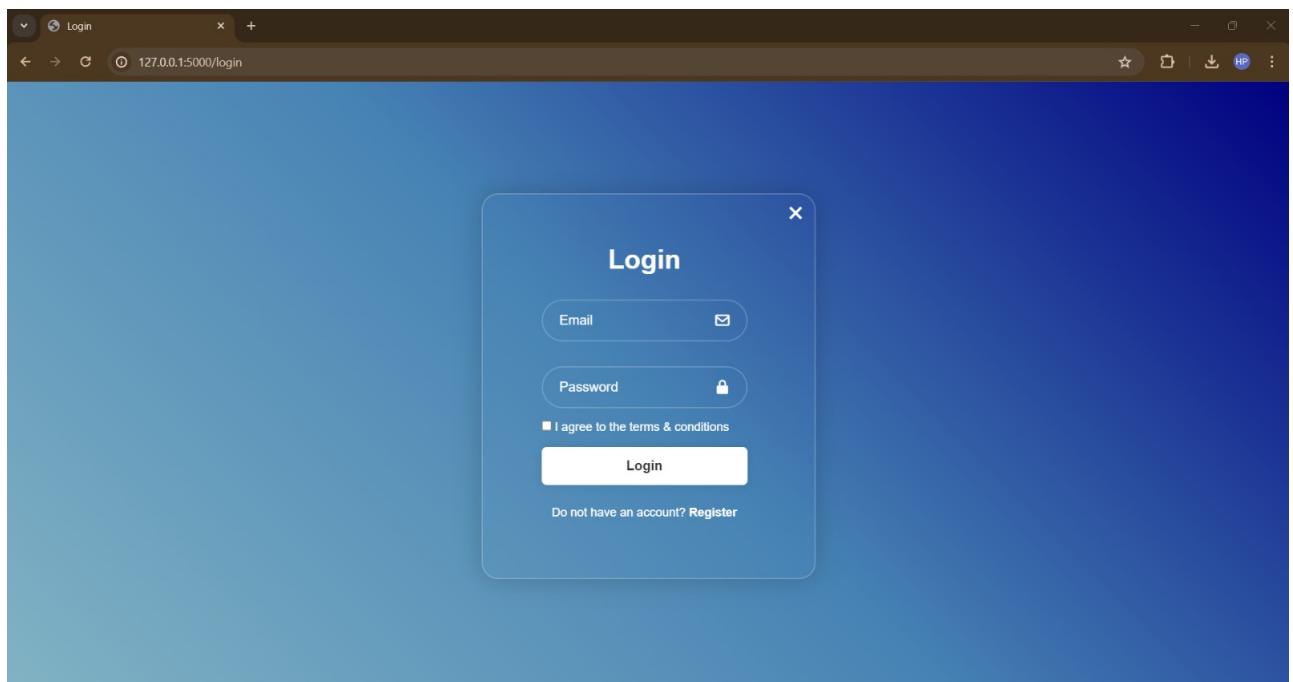


Figure 3.4: Login Page

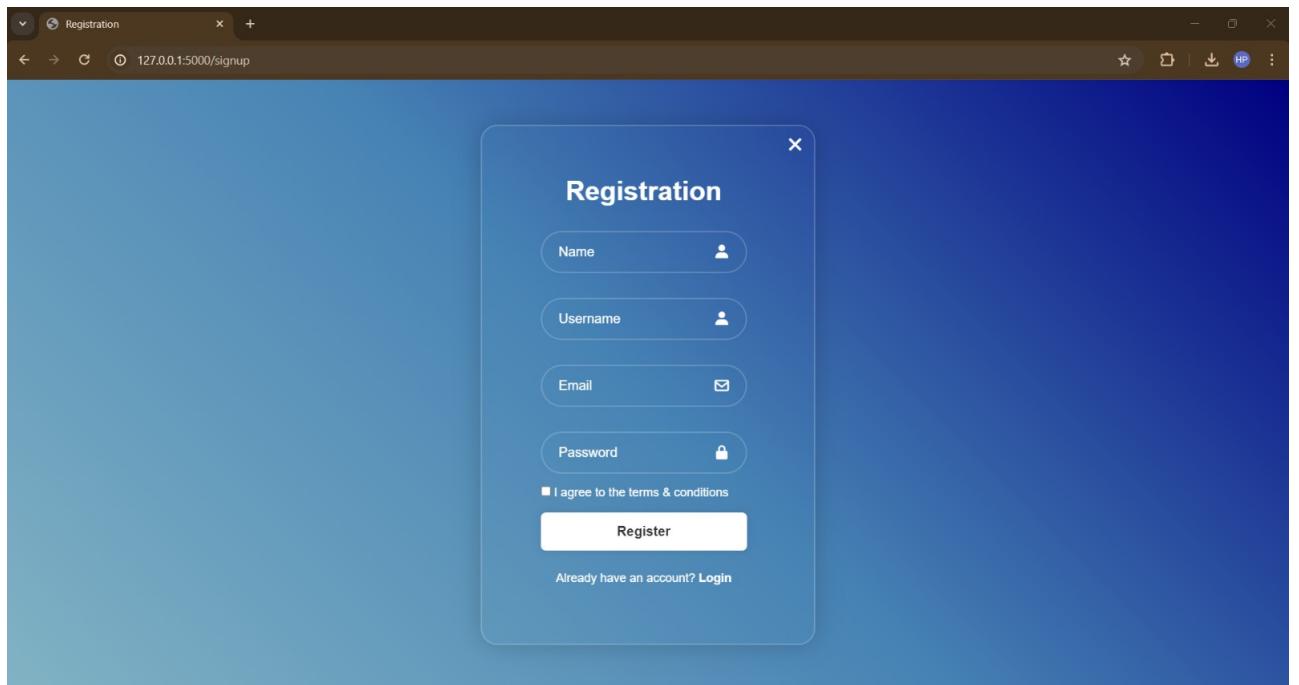


Figure 3.5: Sign Up Page

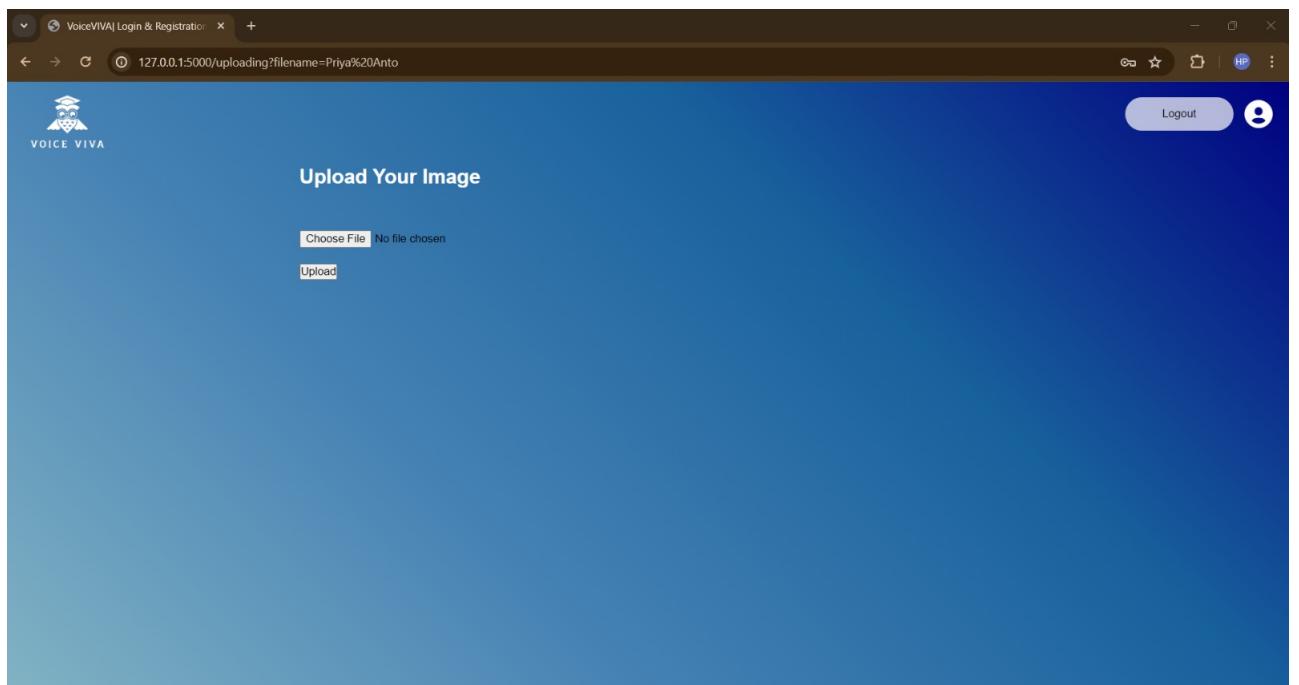


Figure 3.6: To upload image for face recognition after registering the user

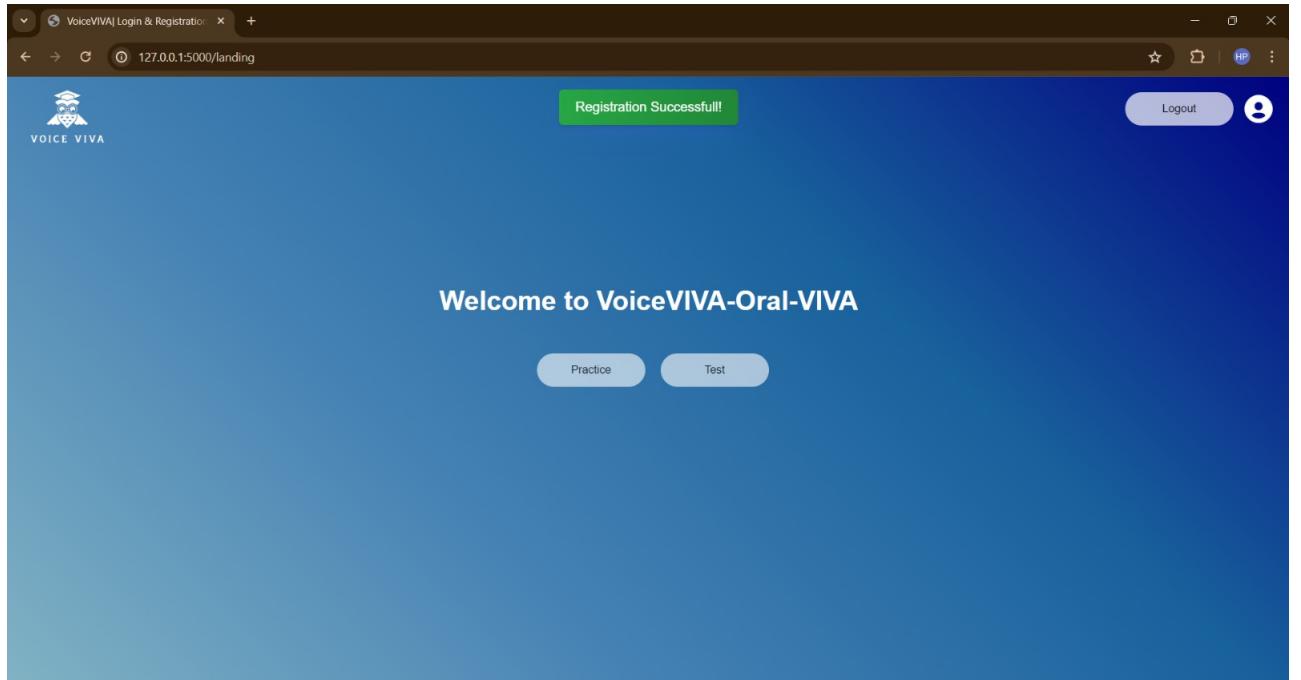


Figure 3.7: Once registration is successful, access to two modes-practice and test

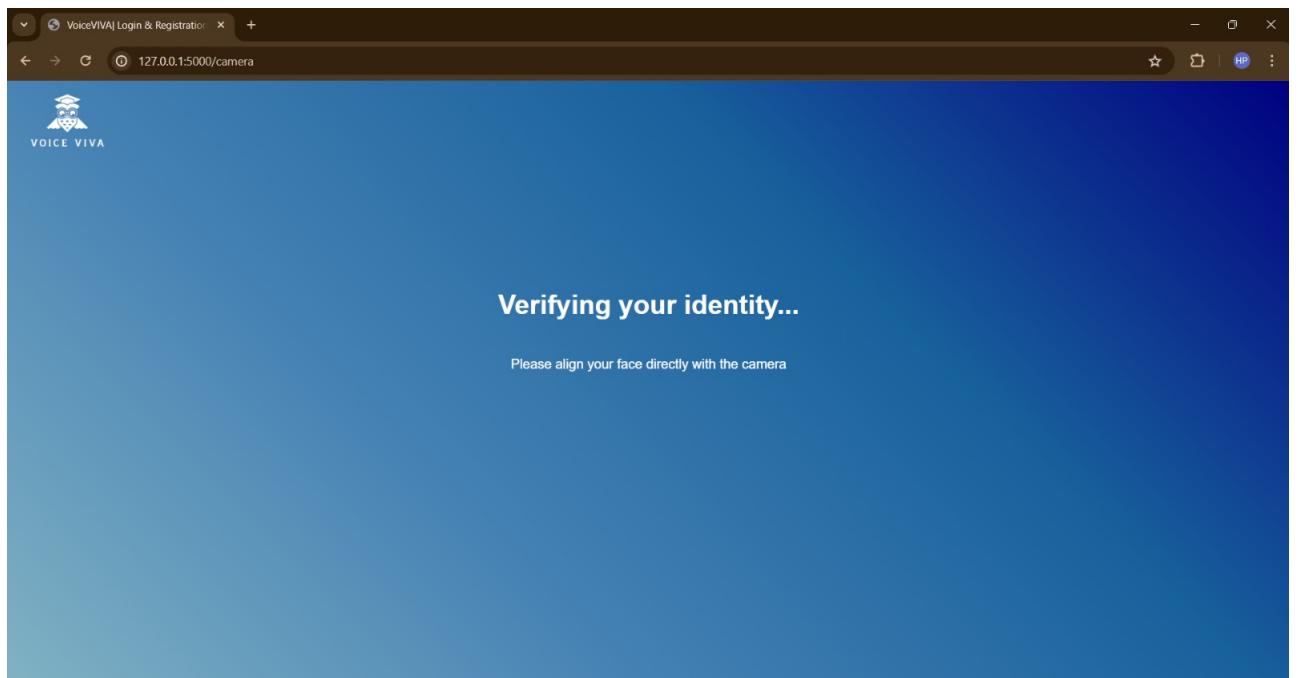


Figure 3.8: Verifying identity of user

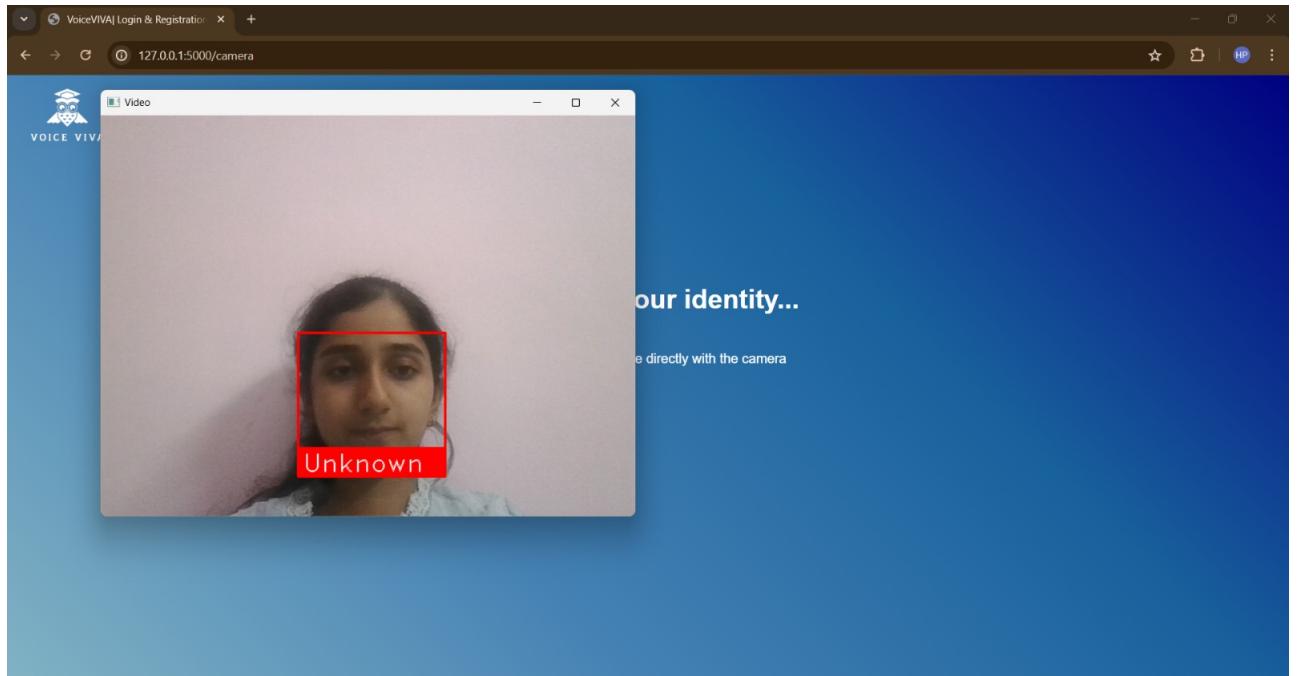


Figure 3.9: Invalid user authentication

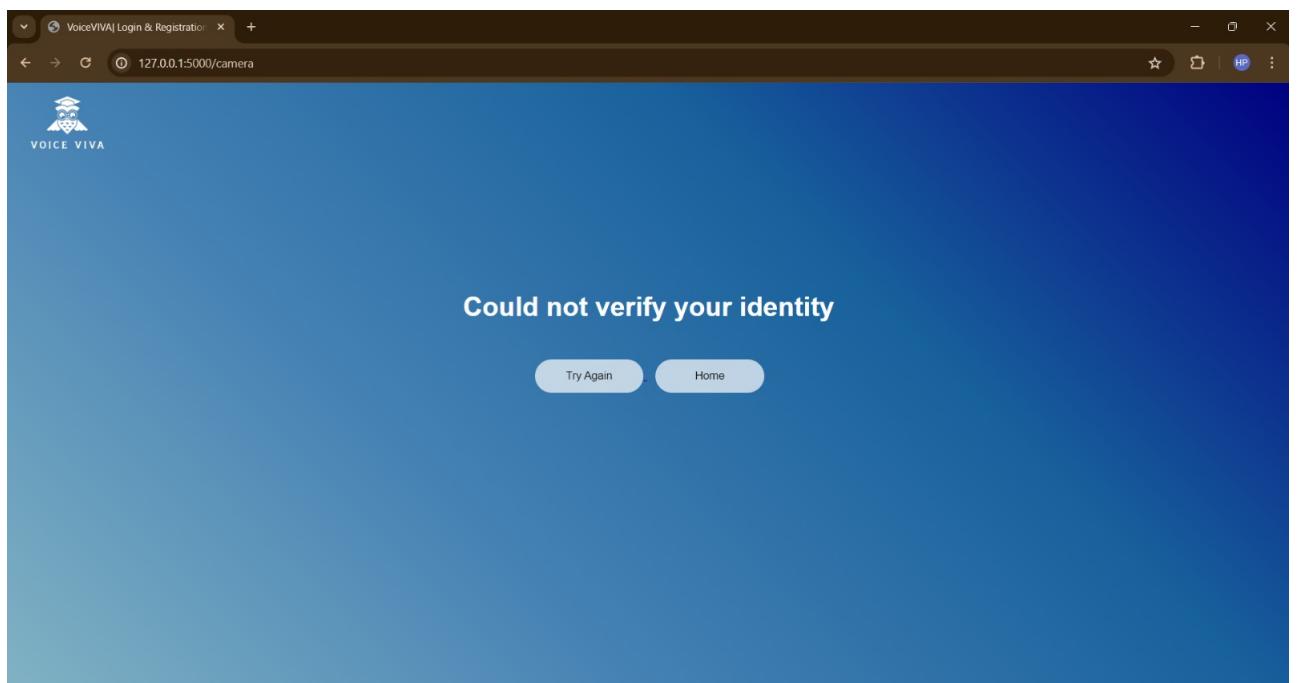


Figure 3.10: Error message on invalid user authentication

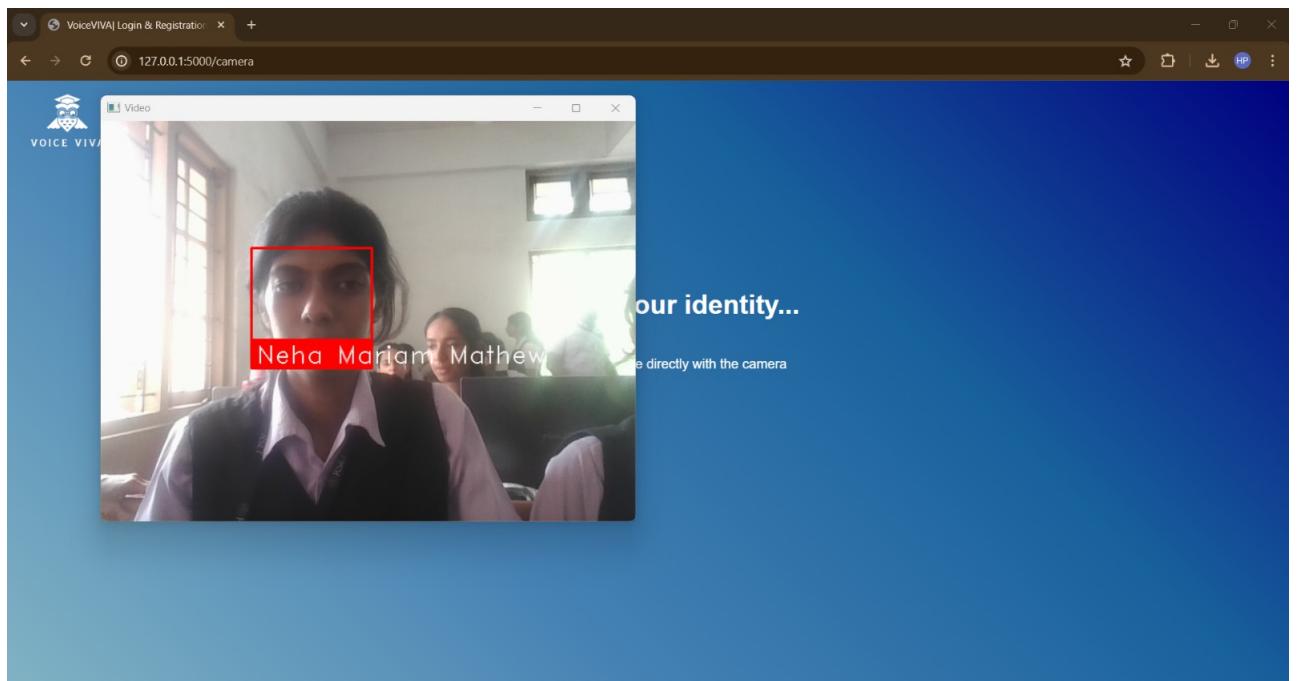


Figure 3.11: Valid user authentication

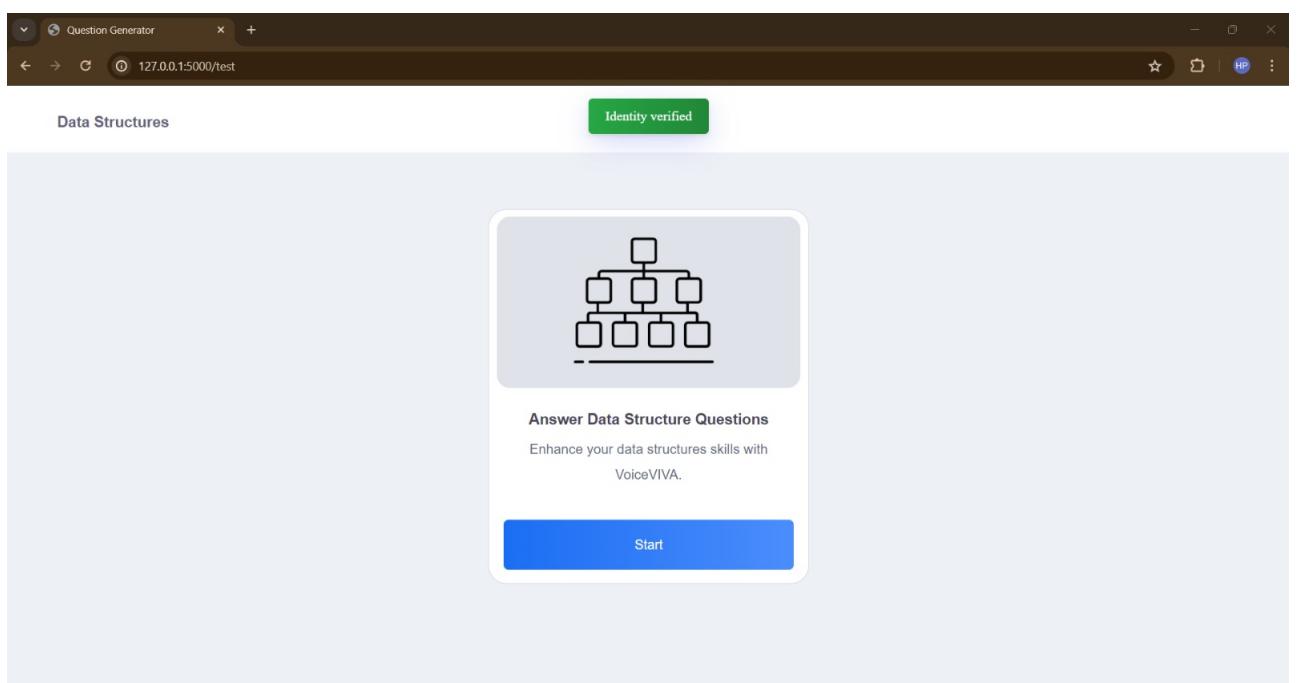


Figure 3.12: On successful user verification-start test

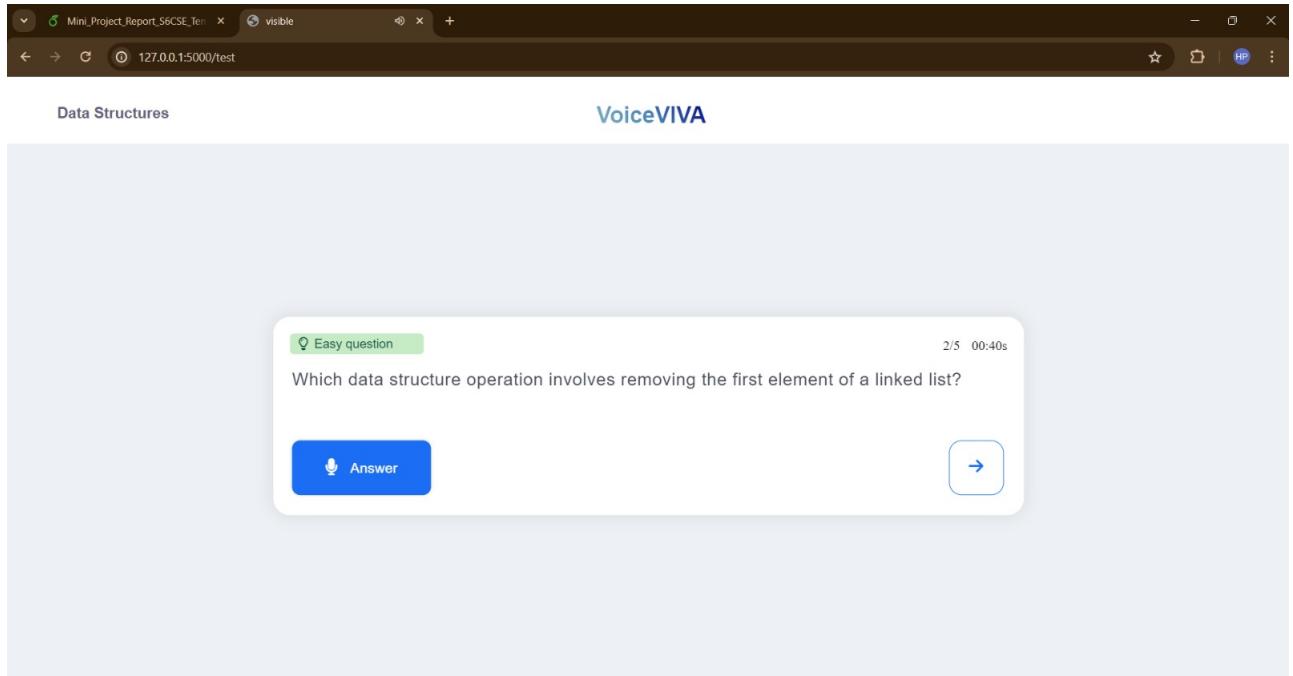


Figure 3.13: Question generation in Test

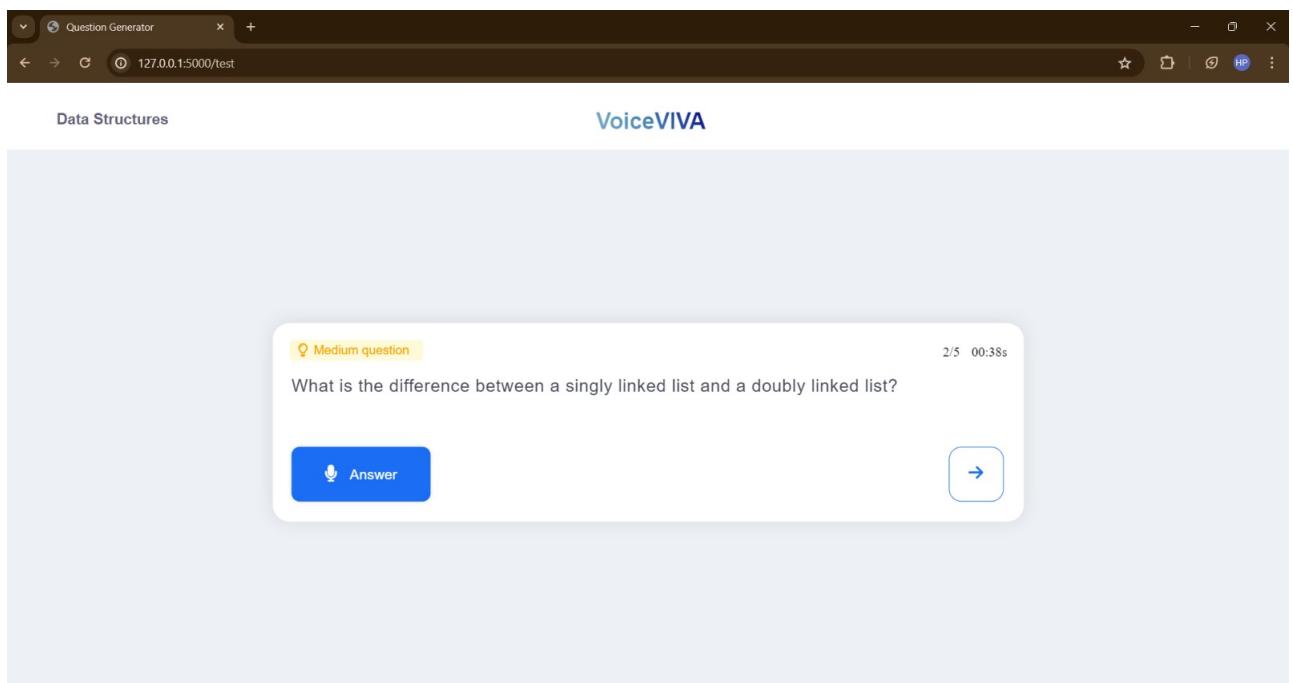


Figure 3.14: Question Generation in test- when the difficulty level is increased

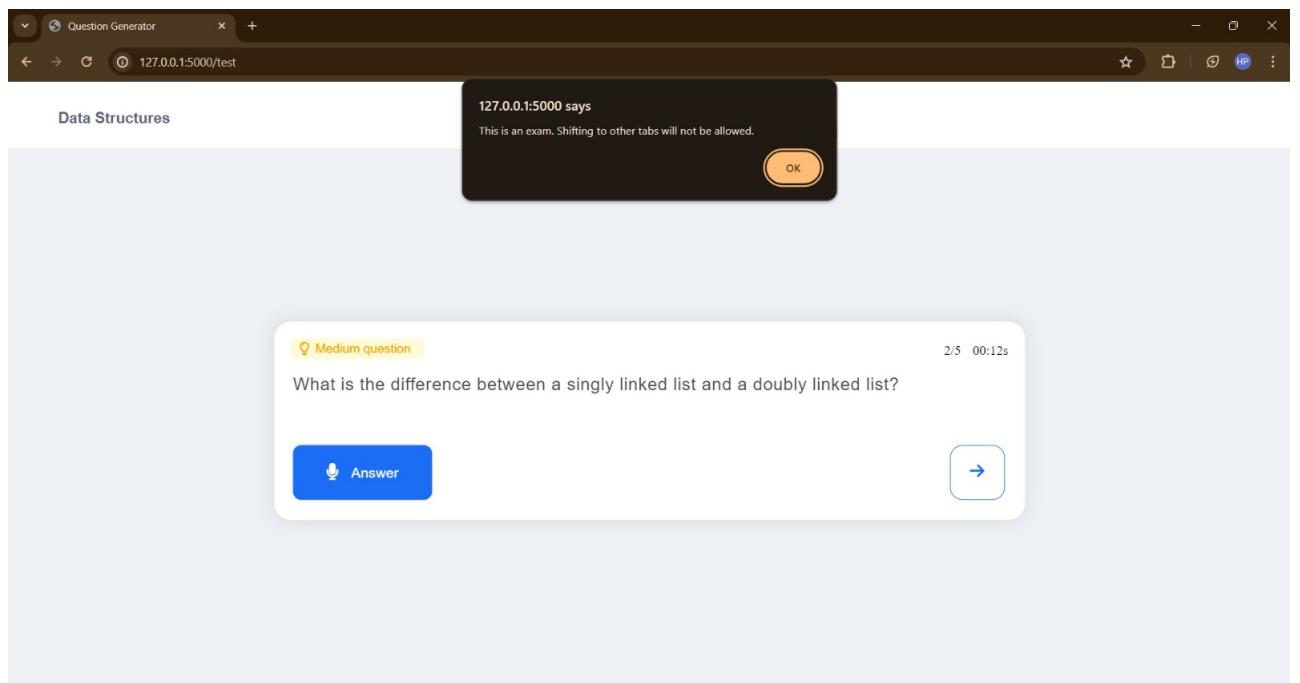


Figure 3.15: Active Tab detection-warning message

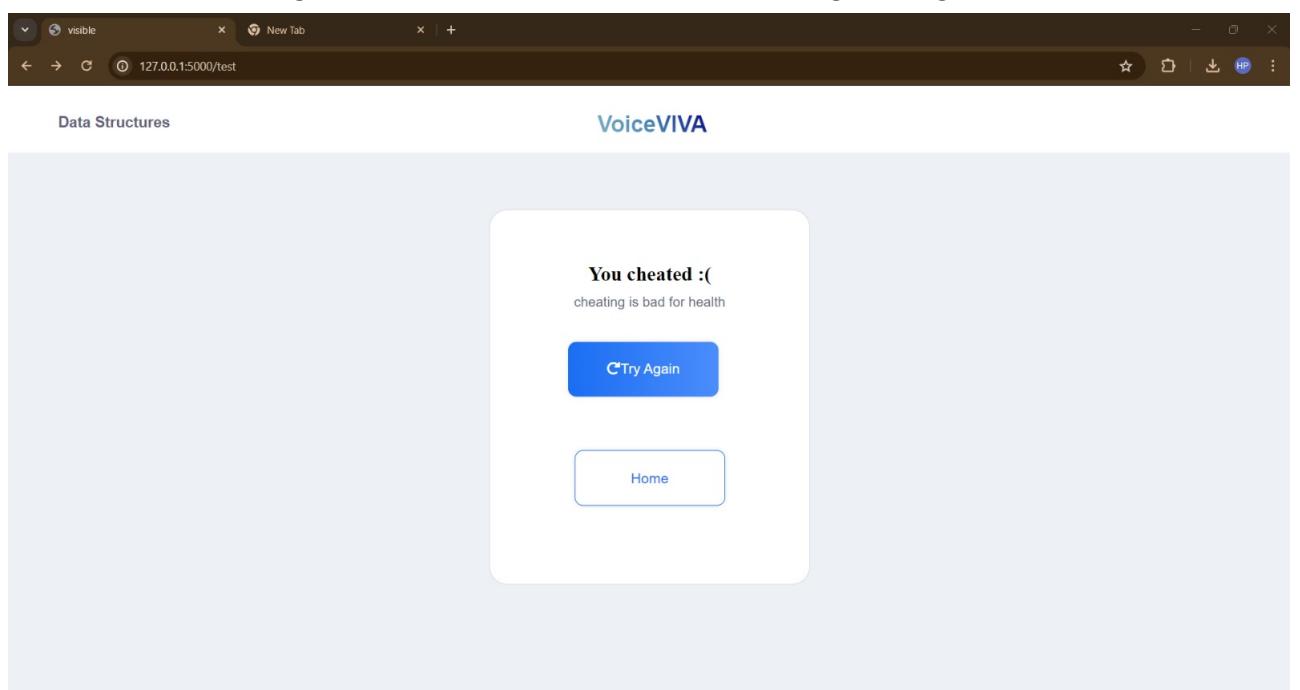


Figure 3.16: Active Tab Detection-test ends

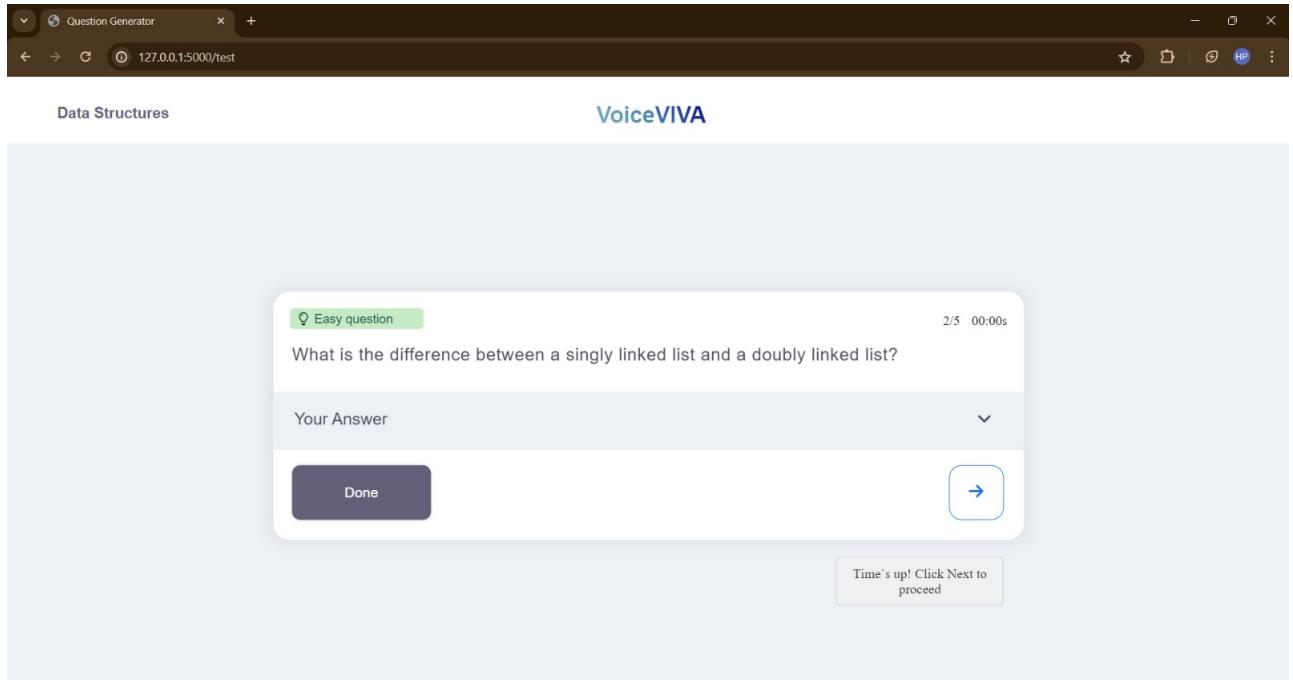


Figure 3.17: Timer elapsed in test

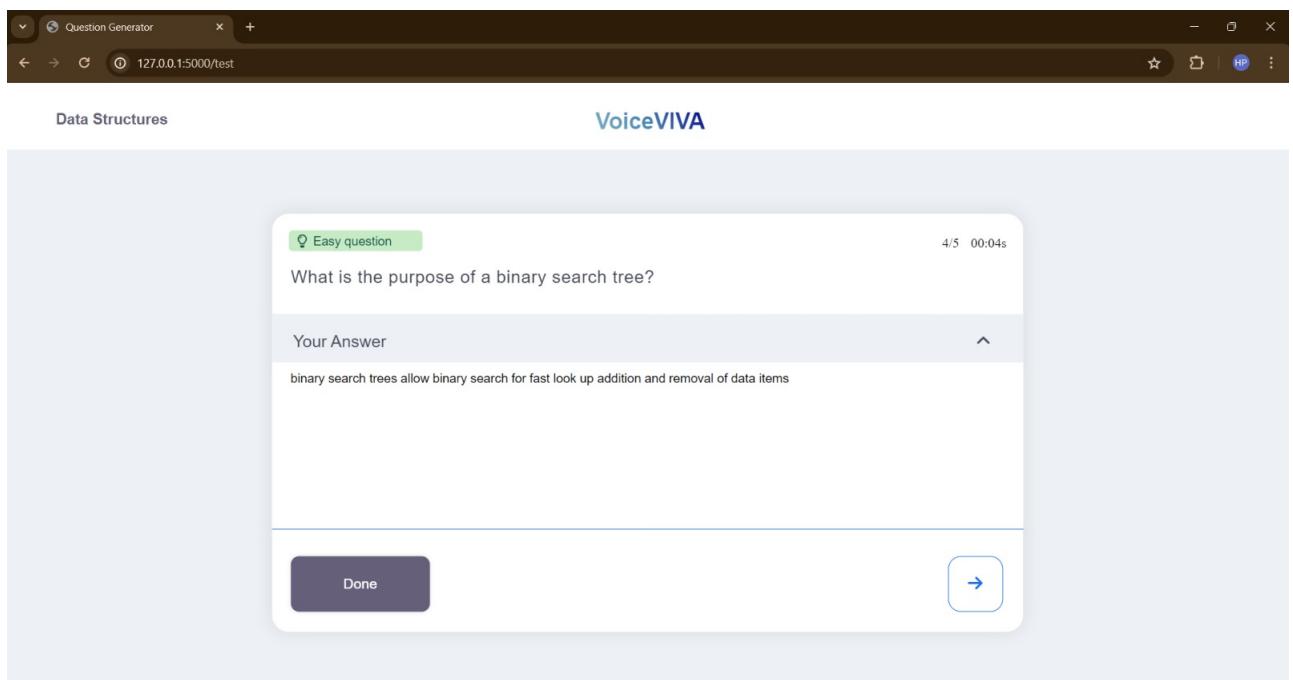


Figure 3.18: Voice to text in test

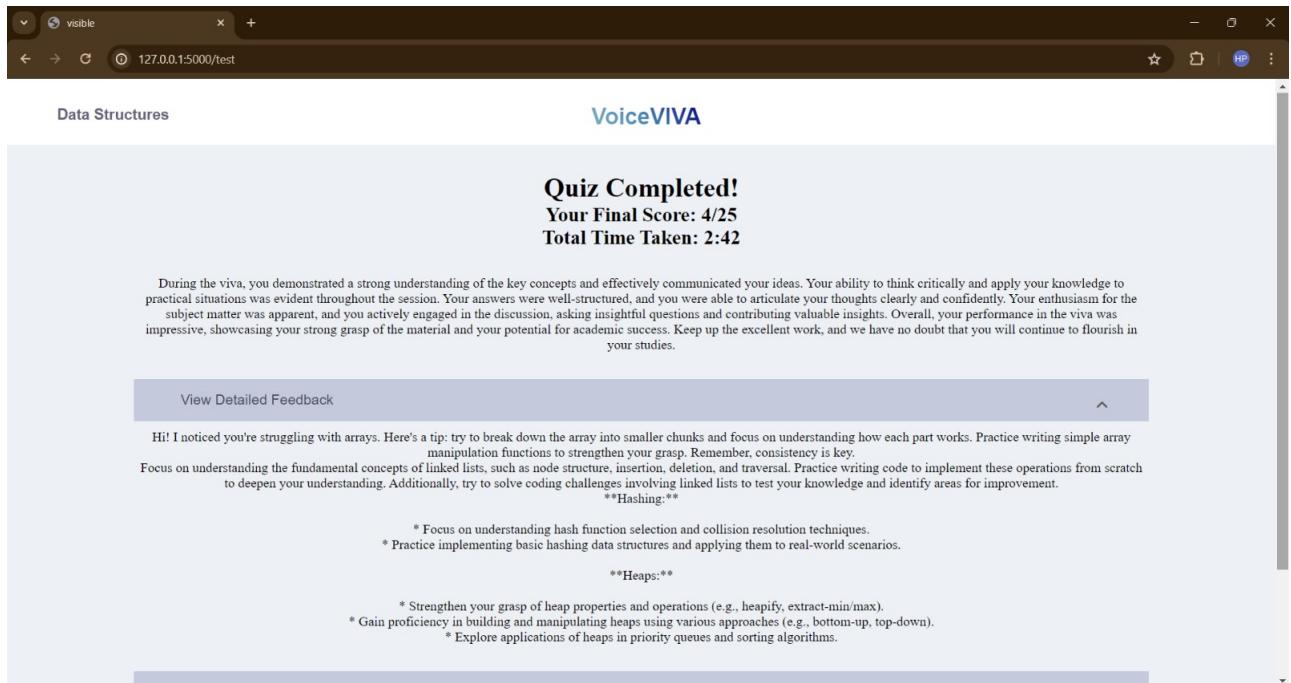


Figure 3.19: End page of test with motivational message, personalized feedback and review questions

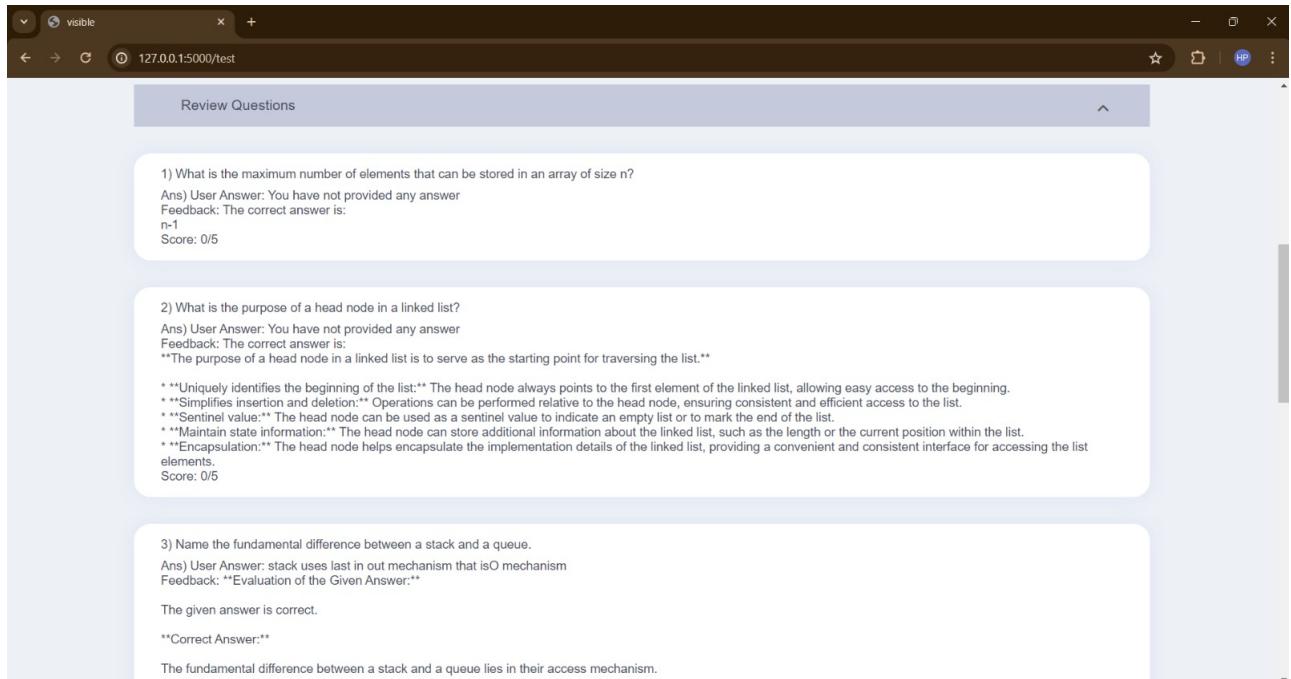


Figure 3.20: Review questions in test

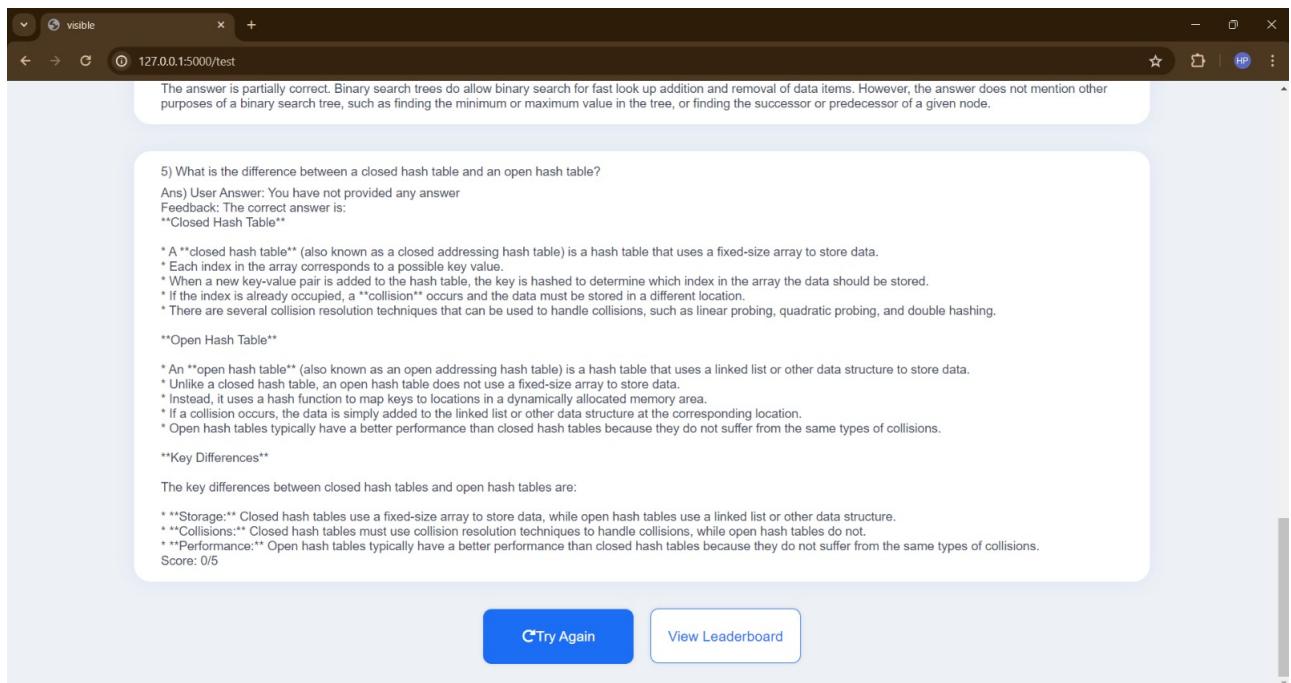


Figure 3.21: A part of end page of test having review question and buttons for try again and leaderboard

RANK	BADGES	NAME	USERNAME	TIME	SCORE
1		Renu Lijo	rinnu	25:12	8
2		Shreya Sunil	shreya6s	1:23	5
3		Neha Mariam Mathew	nehamar	1:56	0
4		Priyo	pam	3:42	0

Figure 3.22: Leaderboard

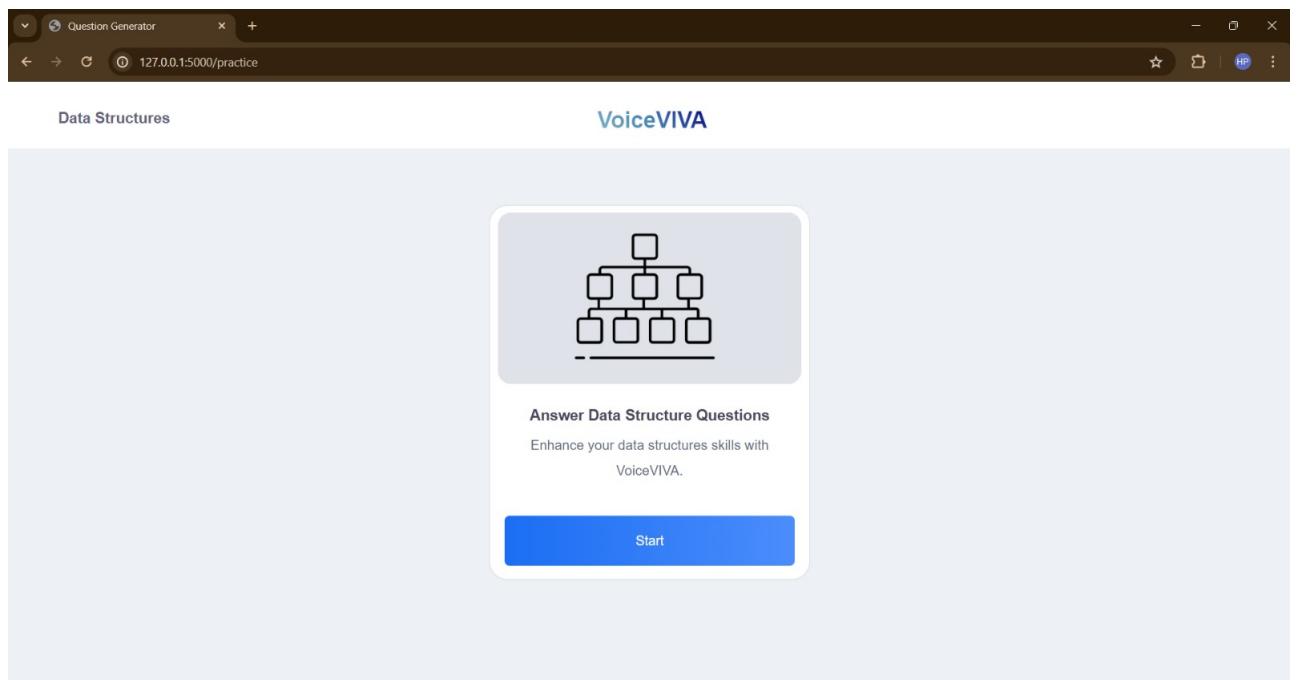


Figure 3.23: Start practice mode

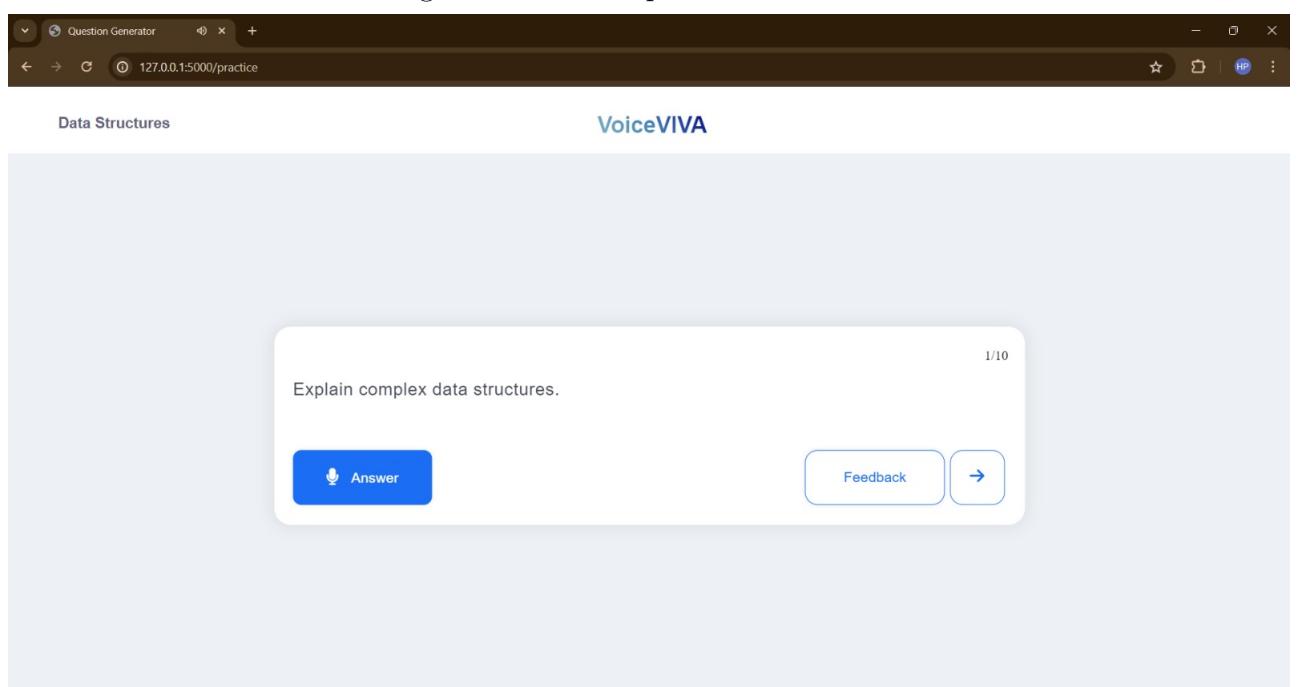


Figure 3.24: Question Generation in practice

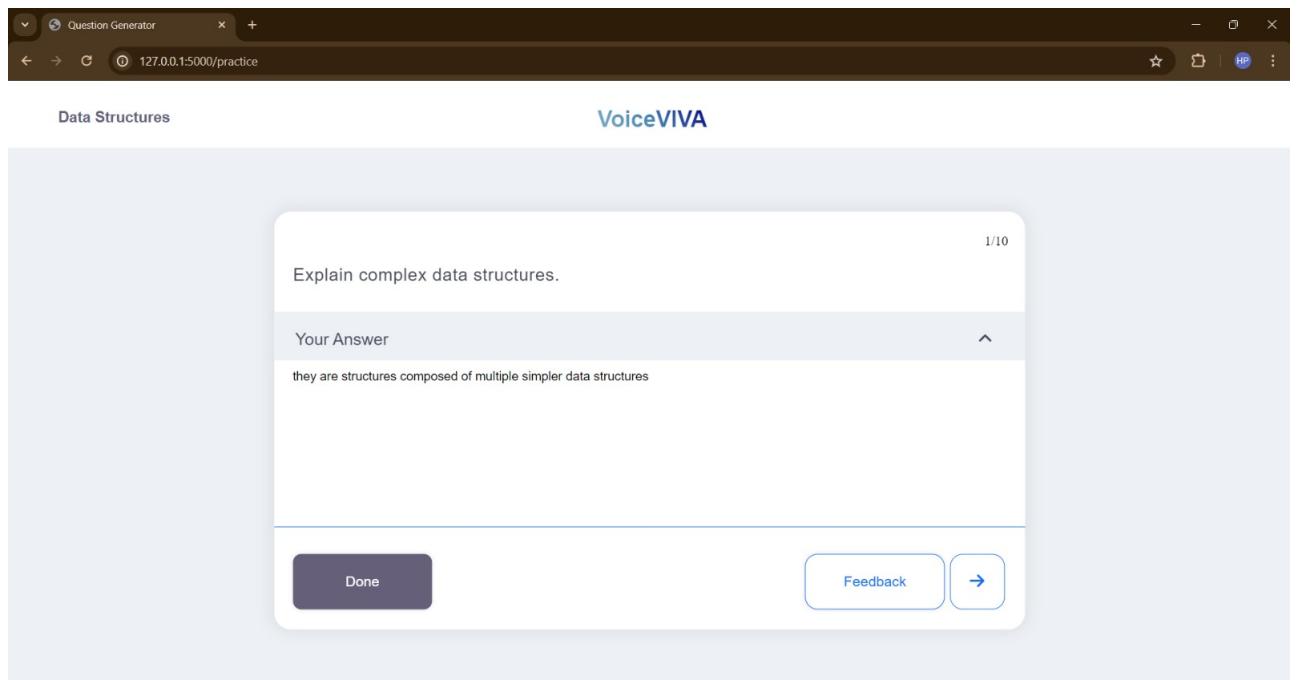


Figure 3.25: Voice to Text in practice

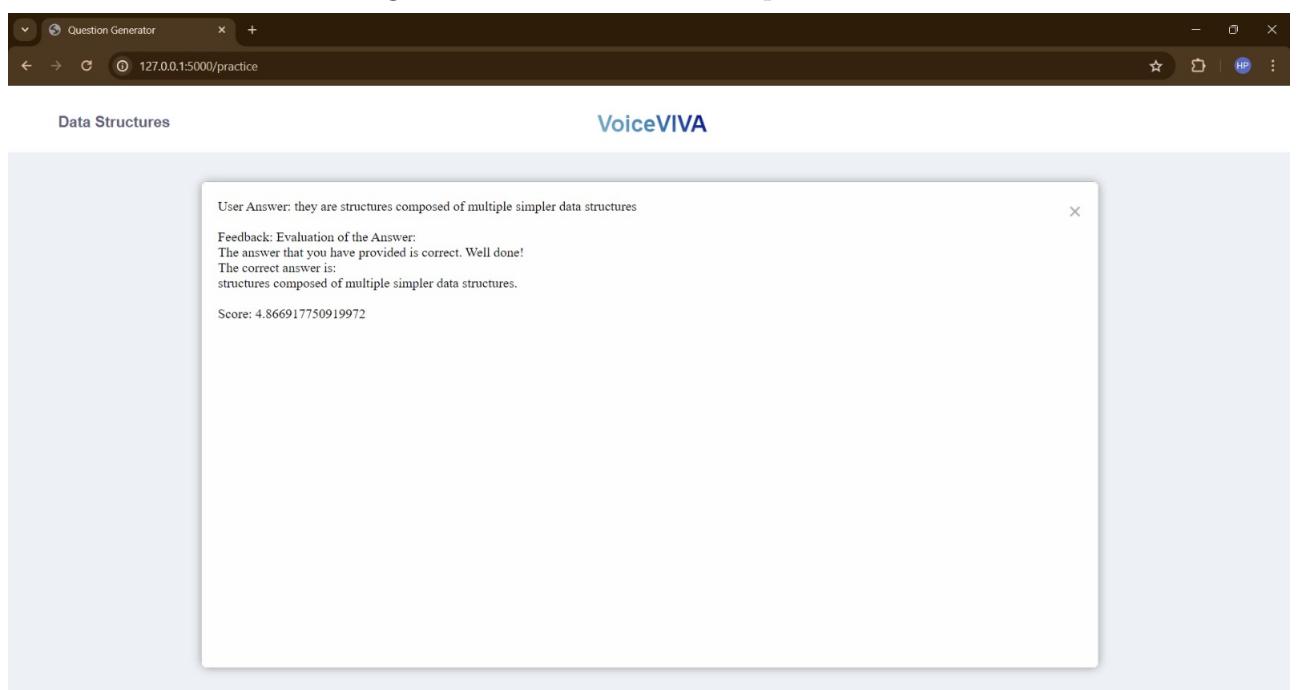


Figure 3.26: Correct Answer Generation, User Answer Evaluation, Feedback and score generation in practice for correct answer

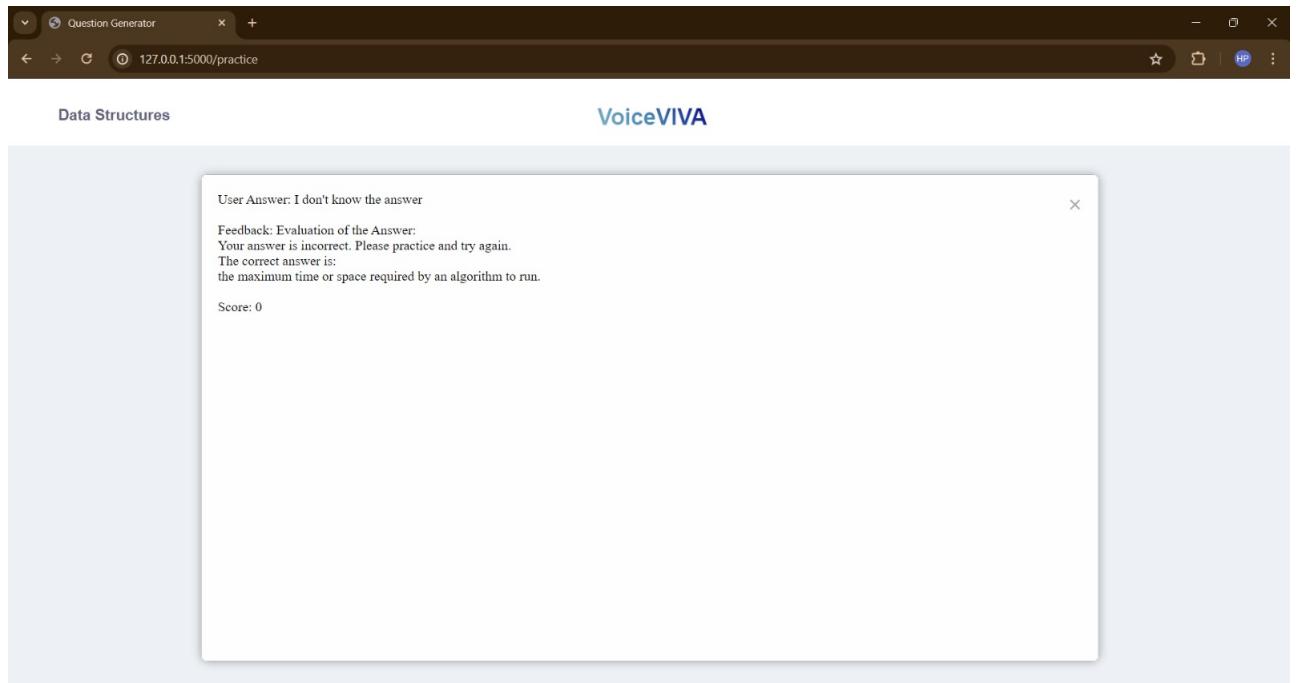


Figure 3.27: Correct Answer Generation, User Answer Evaluation, Feedback and score generation in practice for wrong answer

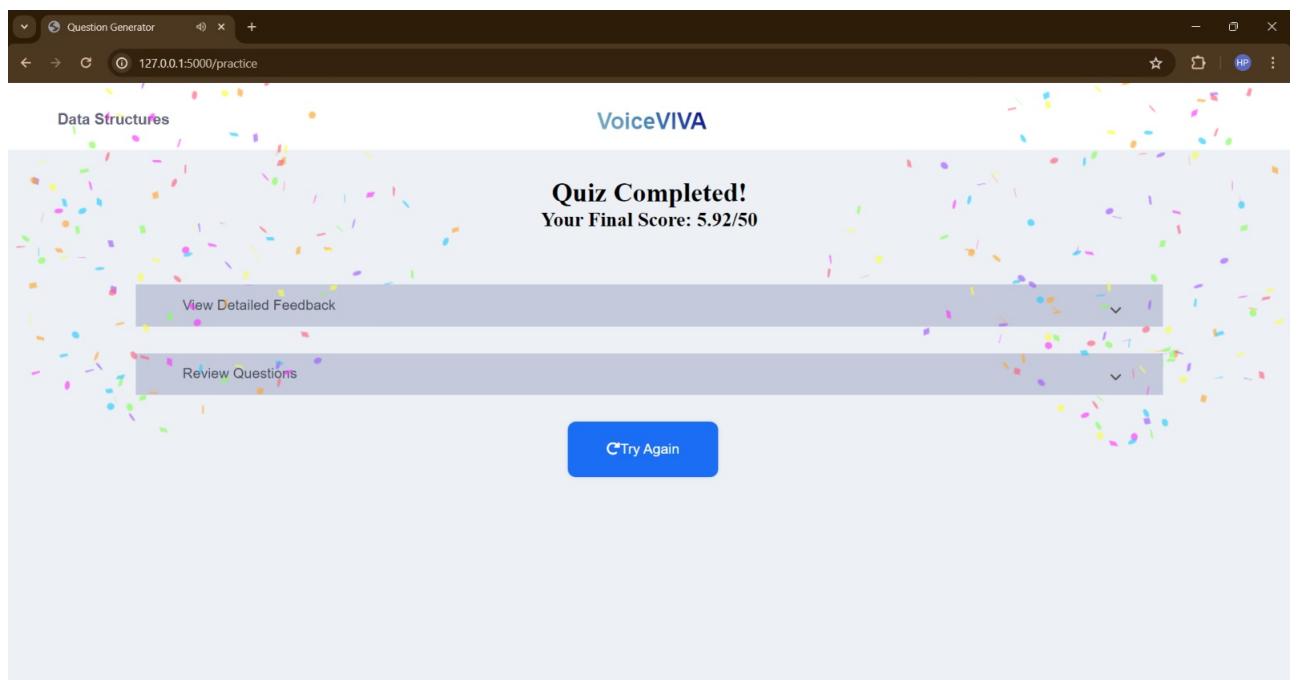


Figure 3.28: End page of practice mode

The screenshot shows a web browser window with the URL `127.0.0.1:5000/practice`. The page content is a detailed feedback report:

- Personalized Feedback on Improving in Stacks and Queues**
- Strengths:**
 - Demonstrated a strong understanding of stack and queue operations (push, pop, enqueue, dequeue).
 - Effectively utilized stacks for function calls and reversing lists.
 - Proficient in applying queues for real-world scenarios, such as job scheduling.
- Areas for Improvement:**
 - Enhance your implementation of complex data structures, such as circular queues and priority queues.
 - Practice solving algorithmic problems that involve stack and queue operations. Focus on optimizing code efficiency and handling edge cases.
 - Explore algorithms utilizing both stacks and queues, like Depth First Search (DFS) and Breadth First Search (BFS).
- Specific Recommendations:**
 - Implement a priority queue data structure using a heap or binary tree.
 - Attempt coding challenges on platforms like LeetCode or HackerRank that feature stack and queue problems.
 - Study the theoretical foundations of stack and queue operations, including proof of correctness and complexity analysis.
- Personalized Feedback on Improving in Trees and Graphs**
- Hello [Your Name].
- Based on your recent work on trees and graphs, I've noticed the following areas where you can improve:
- Understanding graph representation:** Make sure you have a solid grasp of different graph representations (e.g., adjacency list, adjacency matrix) and can convert between them easily.
- Traversals:** Practice various graph traversal algorithms (e.g., DFS, BFS) to enhance your understanding of how to navigate and explore graphs.
- Properties and algorithms:** Study common tree and graph properties (e.g., diameter, height, connected components) and algorithms (e.g., Prim's MST, Kruskal's MST) in detail.
- Proofs:** Practice constructing formal proofs for statements related to trees and graphs. This will improve your logical reasoning and analytical skills.
- Applications:** Explore practical applications of trees and graphs in real-world scenarios, such as network analysis, computer networking, and data structures.

To improve, I recommend:

- Regular practice:** Solve problems and work through examples regularly to reinforce your understanding.
- Research and exploration:** Look up additional resources, such as online tutorials and academic papers, to gain deeper insights.
- Seek help when needed:** Don't hesitate to ask for guidance from your instructor, classmates, or online forums.

Figure 3.29: Detailed personalized report in practice based on your performance in practice, highlighting your strengths and weaknesses

The screenshot shows a web browser window with the URL `127.0.0.1:5000/practice`. The page content is a review questions section:

- 1) Explain complex data structures.**
 Ans) User Answer: They are structures composed of multiple simpler data structures
 Feedback: Evaluation of the Answer:
 The answer that you have provided is correct. Well done!
 The correct answer is:
 structures composed of multiple simpler data structures.
 Score: 4.866917750919972
- 2) Explain the system life cycle.**
 Ans) User Answer: I don't know the answer
 Feedback: Evaluation of the Answer:
 Your answer is incorrect. Please practice and try again.
 The correct answer is:
 the stages through which the system progresses involving design , implementation , operation , maintenance , and replacement.
 Score: 1.056251136725366
- 3) Explain queue.**
 Ans) User Answer: You have not provided any answer
 Feedback: Evaluation of the Answer:
 The correct answer is:
 a linear structure which follows the first in first out , fifo principle.
 Score: 0/0
- 4) What is the dequeue operation?**
 Ans) User Answer: You have not provided any answer
 Feedback: Evaluation of the Answer:

Figure 3.30: Review Questions in practice, allowing to revise the answers to the questions in one glance

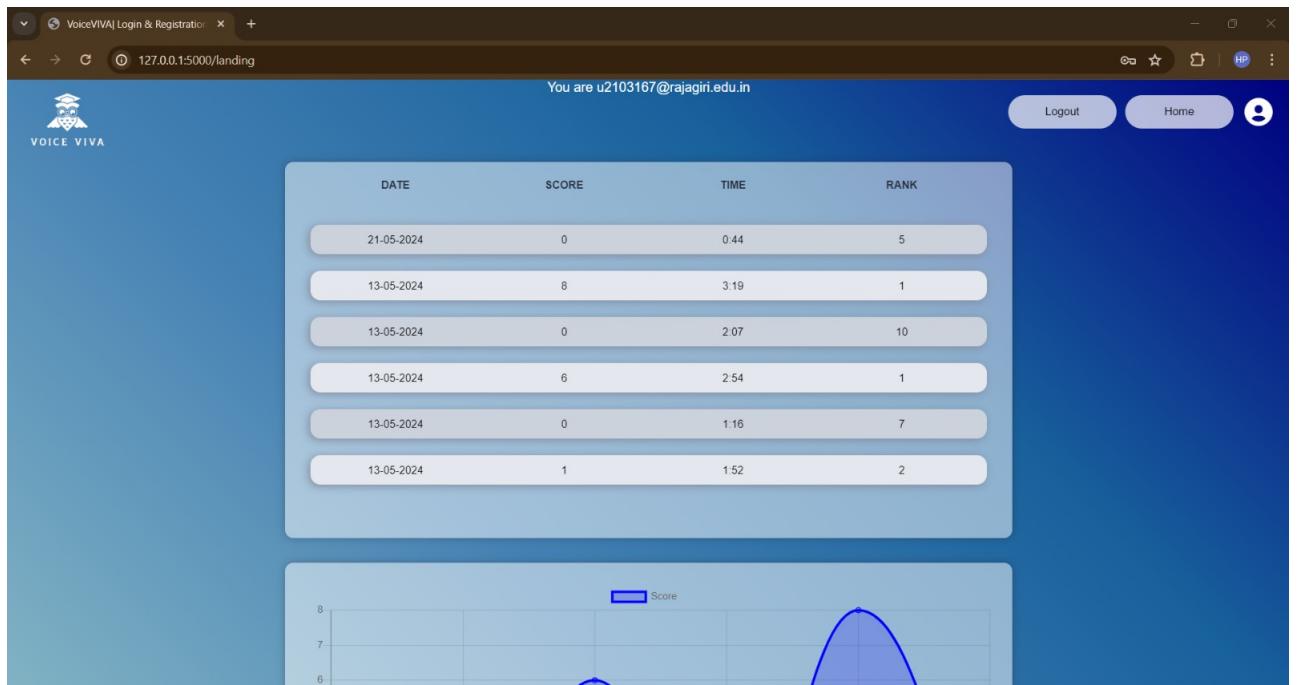


Figure 3.31: Personal profile of an account showing details of your previous attempts

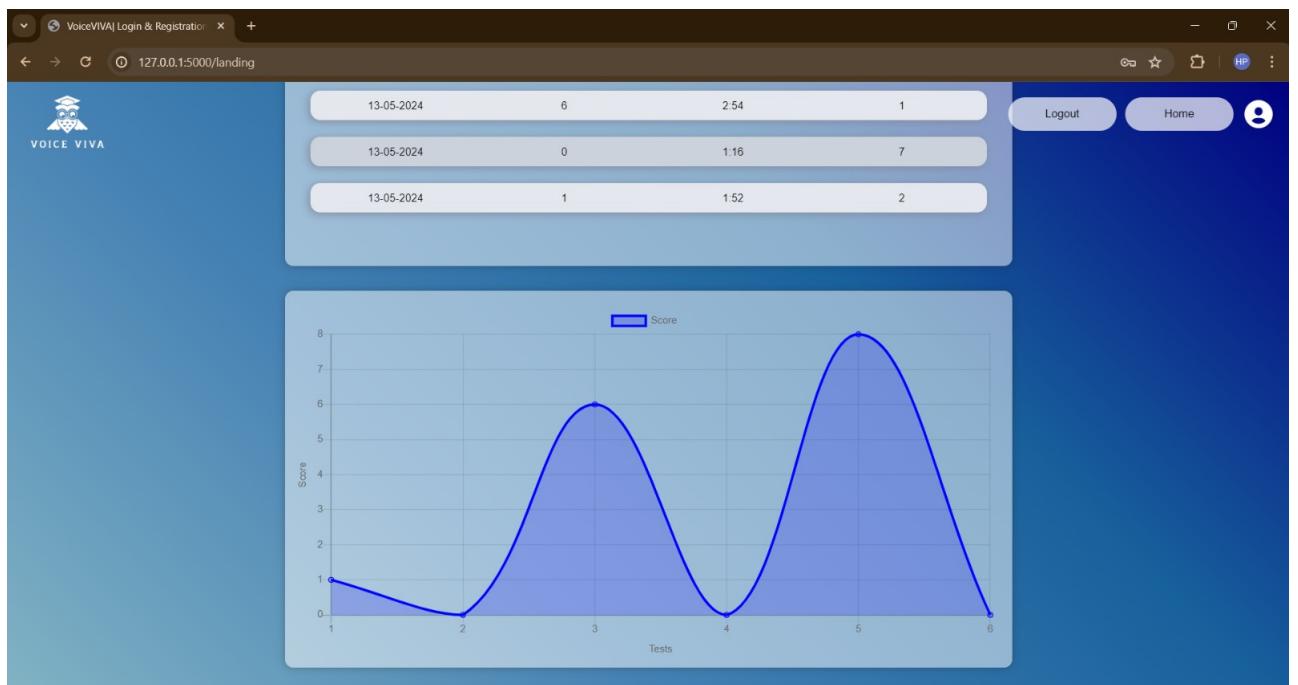


Figure 3.32: Personal profile of an account showing the performance of the user in a graph format

3.5 Database Design

In the development of VoiceVIVA, the choice of database is a critical decision that impacts the performance, scalability, and reliability of the application. Considering the requirements of VoiceVIVA, a NoSQL database like Firebase Realtime Database is a suitable choice.

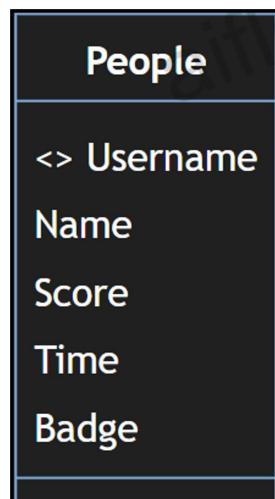


Figure 3.33: People

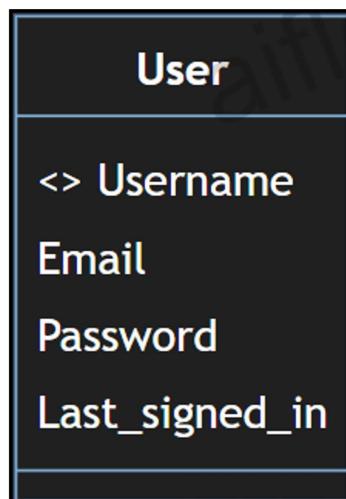


Figure 3.34: User

Reasons for Choosing Firebase:

1. Realtime Updates:

Firebase Realtime Database and Firestore provide real-time synchronization of data

between clients and the database. This feature is crucial for VoiceVIVA, where immediate updates (such as leaderboard rankings or user progress) need to be reflected across all users in real-time.

2. Scalability:

Firebase is highly scalable, allowing VoiceVIVA to handle a growing number of users and data without performance degradation. This scalability is essential for an application that may experience varying levels of usage over time.

3. Offline Support:

Firebase offers offline support, allowing VoiceVIVA to continue functioning even when users are offline. This is beneficial for users in educational environments with inconsistent internet connectivity.

4. Authentication Integration:

Firebase provides robust authentication services, making it easier to implement secure user authentication for VoiceVIVA. This ensures that user data remains protected and accessible only to authorized users.

5. Easy Integration:

Firebase is well-documented and offers easy integration with web applications developed using modern JavaScript frameworks (such as React or Angular), which may be used in the implementation of VoiceVIVA.

6. NoSQL Data Model:

The flexible NoSQL data model of Firebase suits VoiceVIVA's dynamic data requirements, allowing for efficient storage and retrieval of user-related data, assessment results, and leaderboard rankings.

7. Serverless Architecture:

Firebase is a serverless platform, which simplifies the deployment and maintenance of VoiceVIVA. It eliminates the need to manage servers, reducing operational overhead.

—

3.6 Description of Implementation Strategies

1. Voice to Text:

We utilize the Web Speech API, a built-in browser feature. The code enables speech recognition, captures audio input, and converts it to text.

```
1 // Create a new speech recognition object (depending on browser support)
2 const recognition = new webkitSpeechRecognition() || SpeechRecognition();
3 recognition.lang = 'en-US';
4
5 // Allow interim results (show partial transcript while speaking)
6 recognition.interimResults = true;
7 // Enable continuous recognition (keep listening after each phrase)
8 recognition.continuous = true;
9
10 // Start listening for speech input
11 recognition.start();
12 recognition.onresult = function(event) {
13   const transcript = event.results.map(result => result[0].transcript).join('');
14   createParagraph(transcript);
15 };
```

Figure 3.35: Voice to Text Code Snippet

2. Text to Voice:

pyttsx3: Text-to-Speech (converts text to audio) This function uses pyttsx3 to convert question text (text) to audio files (speech).

```
1 # TEXT TO VOICE ( Questions as audio )
2 def generate_question_audio(question_text, question_index):
3     engine = pyttsx3.init()
4     engine.save_to_file(question_text, f'static/question_{question_index}.mp3')
5     engine.runAndWait()
```

Figure 3.36: Text to Voice Code Snippet

3. Question Generation (Practice):

This Python script defines a QuestionExtractor class used in the VoiceVIVA platform to automatically generate questions from a PDF document containing theoretical content. Utilizing libraries like PyMuPDF, NLTK, and Spacy, it processes the text extracted from the PDF, identifies key phrases, ranks them based on relevance, and formulates them into questions. This functionality helps in creating questions which will be utilized in the practice module.

4. Answer Generation (Practice):

```

1 import fitz # PyMuPDF
2 import nltk
3 import spacy
4 from nltk.corpus import stopwords
5 from sklearn.feature_extraction.text import TfidfVectorizer
6 import random
7
8 class QuestionExtractor:
9     def __init__(self, num_questions):
10         # init function
11     def get_questions_from_pdf(self, pdf_path):
12         document = fitz.open(pdf_path)
13         text = "\n".join([page.get_text() for page in document]) # Combine text extraction in one line
14         return self._extract_questions(text)
15
16     def _extract_questions(self, document):
17         sentences = nltk.sent_tokenize(document)
18         for sentence in sentences:
19             self._form_questions(sentence)
20         return self.questions
21
22     def _form_questions(self, sentence):
23         candidate_keywords = [chunk.text for chunk in self.ner_tagger(sentence).noun_chunks]
24         if candidate_keywords:
25             keyword = candidate_keywords[0]
26             question_type = random.randint(1, 3)
27             self.questions.append({"'What is' if question_type == 1 else 'Explain' if question_type == 2 else 'Define'": "s" if keyword.endswith('s') else ""}{keyword.lower(): "s"})
28
29 # Example usage (replace with your actual PDF path)
30 num_questions = 20
31 question_extractor = QuestionExtractor(num_questions)
32 questions = question_extractor.get_questions_from_pdf("path/to/your/pdf.pdf")

```

Figure 3.37: Question Generation(Practice) Code Snippet

Utilizes a pre-trained BERT model to answer a question provided with context. It tokenizes the input, utilizes the model for scoring potential answers, and extracts the most likely answer based on scores.

```

1 from transformers import BertForQuestionAnswering, BertTokenizer
2 import torch
3 import numpy as np
4
5 model = BertForQuestionAnswering.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')
6 tokenizer = BertTokenizer.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')
7
8 def answer(question, context, max_len=512):
9     input_ids = tokenizer.encode(question, context, max_length=max_len, truncation=True)
10    sep_index = input_ids.index(tokenizer.sep_token_id)
11    question_ids, passage_ids = input_ids[:sep_index + 1], input_ids[sep_index + 1:]
12    start_scores, end_scores = model(torch.tensor([question_ids + passage_ids]))[0]
13    start_scores, end_scores = start_scores.detach().cpu().numpy(), end_scores.detach().cpu().numpy()
14    answer_start, answer_end = np.argmax(start_scores), np.argmax(end_scores)
15    answer_tokens = tokenizer.convert_ids_to_tokens(input_ids[answer_start:answer_end + 1])
16    answer = ''.join([t for t in answer_tokens if t not in ['[CLS]', '[SEP]']])
17    return answer if answer_start else "No answer found."
18
19 question = "..." # Generated Question
20 context = "..." # Context
21 print(answer(question, context))

```

Figure 3.38: Answer Generation Code Snippet

5. Score Generation (Practice):

nltk: This library is used for natural language processing tasks such as tokenization (breaking text into words), stopword removal (eliminating common words like "the" and "a"), and lemmatization (converting words to their base form). **scipy spatial distance:** Cosine similarity is a metric used to assess how similar two documents are based on the words they contain. **transformers:** This library is used to load and process a pre-

trained BERT model (BertTokenizer and BertModel). BERT (Bidirectional Encoder Representations from Transformers) is a powerful neural network architecture for natural language understanding tasks. In this context, it is used to generate contextualized word embeddings, which capture the meaning of a word based on its surrounding words.

```

1 #ANSWER CHECKING
2 #KEYWORD CHECKING
3 def process_answers(answer, user_answer):
4     keywords_list = generate_keywords()
5     keywords_generated = find_matched_keywords(answer, keyword_list)
6     keywords_user_answer = find_matched_keywords(user_answer, keyword_list)
7
8     # Call the function to compare user_answer with matched_keywords
9     matched_keywords_in_user_answer = find_matching_keywords(keywords_generated, keywords_user_answer)
10
11    # Calculate the percentage of matched keywords in user's answer
12    # KEYWORD SCORE
13    if keywords_generated:
14        keyword_score = (len(matched_keywords_in_user_answer) / len(keywords_generated)) *5
15
16    # Call function to generate overall score
17    overall_score = score_generation()
18
19
20 # SYNONYM CHECKING
21 def score_generation(answer2, user_answer2, keyword_score, p, answer, user_answer):
22     # Finding the similar words between user answer and generated answer
23     similar_words = [word1 for word1 in set(preprocess_text(similar(answer2))) for word2 in set(preprocess_text(similar(user_answer2)))
24     | | | | | if max(s1.path_similarity(s2) or 0) for s1 in wn.synsets(word1) for s2 in wn.synsets(word2)) >= 0.7]
25
26     similarity_score = compare_similarity(answer2, similar_words)
27     partial_score = (keyword_score + similarity_score) / p
28     cosine_similarity = cosineSimilarity.cosine_similarity(answer, user_answer)
29     overall_score = (partial_score + cosine_similarity) / 2
30     return overall_score
31
32 #COSINE SIMILARITY
33 def cosine_similarity(sentence1, sentence2):
34     tokens1, tokens2 = BertTokenizer.from_pretrained('bert-base-uncased')(sentence1, return_tensors='pt', max_length=128, truncation=True, padding=True),
35     BertTokenizer.from_pretrained('bert-base-uncased')(sentence2, return_tensors='pt', max_length=128, truncation=True, padding=True)
36     embeddings1 = BertModel.from_pretrained('bert-base-uncased')(**tokens1).last_hidden_state.mean(dim=1).squeeze().numpy()
37     embeddings2 = BertModel.from_pretrained('bert-base-uncased')(**tokens2).last_hidden_state.mean(dim=1).squeeze().numpy()
38
39     similarity_score = 1 - cosine(embeddings1, embeddings2)
40
41     # Consider negation and antonyms
42     if (has_negation(sentence1) != has_negation(sentence2)) or any([has_antonym(phrase1, phrase2) for phrase1 in set(sentence1.lower().split())
43     for phrase2 in set(sentence2.lower().split())]):
44         similarity_score = 0
45
46     return similarity_score * 5

```

Figure 3.39: Score Generation (Practice) Code Snippet

6. Answer Evaluation (Practice):

Answer is evaluated and feedback is generated according to the generated score

```

1 feedback_message=['The answer that you have provided is correct. Well done!', 'Not bad! Your answer is close to the correct answer.', 
2 | | | | | 'You need to improve. Your answer is partially correct.', 'Your answer is incorrect. Please practice and try again.']
3 if overall_score>4.8:
4     feedback=feedback_message[0]
5     elif overall_score>4:
6         feedback=feedback_message[1]
7     elif overall_score>3:
8         feedback=feedback_message[2]
9     elif overall_score<3:
10        feedback=feedback_message[3]

```

Figure 3.40: Answer Evaluation (Practice) Code Snippet

7. Question Generation (Test):

Method: The generate question route in app.py dynamically generates questions using the Gemini API based on predefined topics and difficulty levels. The generated questions are then presented to the user for answering.

```

1  @test_bp.route('/generate_question')
2  def generate_question():
3      global current_question_index, k, generating_question, d, quiz_start_time, g_answer, gemini_answer
4      if quiz_start_time is None:
5          quiz_start_time = time.time() # Record the current time as the start time of the quiz
6      if not generating_question and current_question_index < 5:
7          generating_question = True
8          if '*' not in globals():
9              k = 0
10             d = 0
11         try:
12             question_response = make_gemini_request("Ask an "+diff[d]+" one line question(without answer) about data
13             structures(don't ask time complexity questions) only from the topic "+topics[k])
14             print(topics[k],diff[d])
15             k += 1
16             try:
17                 question_text = question_response['candidates'][0]['content']['parts'][0]['text']
18                 questions.append(question_text)
19                 correct=make_gemini_request(f"Generate answer for the question: {question_text}")
20                 if correct and 'candidates' in correct:
21                     gemini_answer = correct['candidates'][0]['content']['parts'][0]['text']
22                     correct_answer.append(gemini_answer)
23                     current_question_index += 1
24                     generate_question_audio(question_text, current_question_index)
25                     generating_question = False
26                     return jsonify({'question': question_text,'difficulty': diff[d]}) # Return the question text
27                 except IndexError:
28                     generating_question = False
29                     return jsonify({'error': 'Unable to access question text'})
30                 except requests.exceptions.RequestException as e:
31                     generating_question = False
32                     return jsonify({'error': 'Unable to generate question'})
33             else:
34                 return jsonify({'error': 'Question generation in progress or quiz completed'})
35         def adjust_difficulty():
36             global d
37             if mark<=2:
38                 if d==1:
39                     d=0
40                 elif d==2:
41                     d=1
42             elif mark>3:
43                 if d==0:
44                     d=1
45                 elif d==1:
46                     d=2
47 GENERATE QUESTION BASED ON DIFFICULTY

```

Figure 3.41: Question Generation (Test) Code Snippet

8. Answer Evaluation (Test):

Library: The project employs Google's Gemini API for natural language processing.

Method: In the evaluate answer route of app.py, the Gemini API is utilized to evaluate the correctness of user answers and assign scores. The API is called with the user's answer and the corresponding question prompt for evaluation.

9. Flask Integration:

Library: The project utilizes the Flask web framework for backend implementation.

Method: Routes are defined in app.py, test.py and practice.py to handle various func-

```

1 # Route to evaluate user answer
2 @test_bp.route('/evaluate_answer', methods=['POST'])
3 def evaluate_answer():
4     global current_question_index, sum, mark, question_answers, g_answer, gemini_answer
5     data = request.get_json()
6     user_answer = data.get('user_answer')
7
8     # Set user answer to an empty string if it's None
9     if user_answer is None:
10         user_answer = ""
11
12     evaluation_response = make_gemini_request(f"Evaluate the correctness of following answer: {user_answer} for the question: {questions[current_question_index - 1]}")
13                                         # Provide the correct answer for the {questions[current_question_index - 1]}
14     evaluation_score = make_gemini_request(f"Assign a score out of 5 for the following answer: {user_answer} to the question: {questions[current_question_index - 1]}"
15                                         # based on its correctness")
16
17     if not evaluation_response or 'candidates' not in evaluation_response:
18         return jsonify({'error': 'Unable to evaluate answer'})
19     if not evaluation_score or 'candidates' not in evaluation_score:
20         return jsonify({'error': 'Unable to evaluate answer'})
21
22     score = evaluation_score['candidates'][0]['content']['parts'][0]['text']
23     feedback = evaluation_response['candidates'][0]['content']['parts'][0]['text']
24
25     entry = [questions[current_question_index - 1], {'user_answer': user_answer, 'feedback': feedback, 'score': score}]
26     question_answers[current_question_index - 1] = entry
27
28     if 'sum' not in globals():
29         sum = 0
30         mark=0
31
32     for x in score:           #extracting the first digit bcoz that will be the score
33         if x.isdigit():
34             break
35
36     sum+=int(x)
37     mark=int(x)
38     print(mark)
39     adjust_difficulty()
40     return jsonify({'feedback': feedback, 'score': sum, 'mark': mark})
41
42 EVALUATE

```

Figure 3.42: Answer Evaluation (Test) Code Snippet

tionalities such as question generation, answer evaluation, and feedback generation. Flask enables handling HTTP requests and responses efficiently.

10. Front end:

HTML, CSS, and JavaScript are used for building the frontend components of the website, such as user interfaces, buttons, and event handling.

11. Firebase:

For establishing a connection with Firebase in the project, we utilize the Firebase JavaScript SDK. This SDK provides methods for initializing Firebase, accessing the Firebase Realtime Database, and performing various database operations. Through this SDK, we can access the information needed for the leaderboard, such as user scores, names, and other relevant details stored in the Firebase Realtime Database.

12. Leaderboard:

```

1  from flask import Blueprint
2  # ... other imports (from practice.py and test.py)
3  # Register Blueprints
4  app.register_blueprint(practice_bp)
5  app.register_blueprint(test_bp)
6  @app.route('/')
7  def opening():
8      | # Landing page route
9      @app.route('/signup')
10     def signup():
11         | # Signup page route
12         @app.route('/signup', methods=['POST'])
13     def signup_post():
14         | # Handles signup form submission
15     @app.route('/login')
16     def login():
17         | # Login page route
18         @app.route('/login', methods=['POST'])
19     def login_post():
20         | # Handles login form submission
21     @app.route('/landing')
22     def landing():
23         | # Main landing page route after login
24     @app.route('/logout')
25     def logout():
26         | # Logout route
27     @app.route('/camera')
28     def camera():
29         | # Camera page route
30     @app.route('/uploading')
31     def upload_form():
32         | # Upload form route
33     @app.route('/upload', methods=['POST'])
34     def upload_file():
35         | # Handles image upload
36     @app.route('/download')
37     def download_file():
38         | # Handles image download
39     # Removed commented-out route for brevity
40     @app.route('/leaderBoard')
41     def leaderboard():
42         | # Leaderboard page route

```

Figure 3.43: Flask Integration Code Snippet

```

1  Snippet:
2  <link rel="stylesheet" type="text/css" href="static/styles.css">
3  <script type="module">
4  | // JavaScript code for frontend functionality
5  </script>

```

Figure 3.44: Front end Code Snippet

```

1  Code Snippet:
2  // Import the Firebase SDK and Firebase Database module
3  import { initializeApp } from "https://www.gstatic.com/firebasejs/10.10.0/firebase-app.js";
4  import { getDatabase } from "https://www.gstatic.com/firebasejs/10.10.0.firebaseio.js";
5
6  // Firebase configuration object obtained from Firebase console
7  const firebaseConfig = {
8      apiKey: "YOUR_API_KEY",
9      authDomain: "YOUR_AUTH_DOMAIN",
10     databaseURL: "YOUR_DATABASE_URL",
11     projectId: "YOUR_PROJECT_ID",
12     storageBucket: "YOUR_STORAGE_BUCKET",
13     messagingSenderId: "YOUR_MESSAGING_SENDER_ID",
14     appId: "YOUR_APP_ID"
15   };
16
17  // Initialize Firebase
18  const firebaseApp = initializeApp(firebaseConfig);
19
20  // Get a reference to the Firebase Realtime Database service
21  const db = getDatabase(firebaseApp);|

```

Figure 3.45: Firebase Code Snippet

Fetches data (People) from Firebase, ordered by Score . Reverses order to get the highest score First. Sorts data by user completion time. Finally, displays the ranked data in a leaderboard table.

```

1 <script>
2 // Firebase initialization (assuming configured elsewhere)
3 const db = getDatabase(app);
4
5 fetchAndDisplayData();
6
7 function fetchAndDisplayData() {
8 // fetching data order by Score
9   get(query(ref(db, "People"), orderByChild("Score").limitToLast(10)))
10   .then(snapshot => {
11     const dataArray = snapshot.val() || [];
12     const reversedArray = dataArray.reverse(); // Reverse the data
13     dbSort(reversedArray); // Sort the data based on time
14     displayData(reversedArray);
15   })
16   .catch(error => console.error(error));
17 }
18
19 function displayData(dataArray) {
20   const tableRows = document.querySelector(".tableRows");
21   let rank = 1;
22   dataArray.forEach(user => {
23     const row = createRow(rank++, user, email);
24     tableRows.appendChild(row);
25   });
26 }
27
28 function createRow(rank, user, email) {
29   // ... (row creation logic with badge checks and formatting)
30   return row;
31 }
32
33 // Your existing dbSort function for time-based sorting
34 function dbSort(snapshot) {
35   // ... (implementation of bubble sort based on Time)
36 }
37 </script>
```

Figure 3.46: Leaderboard Code Snippet

3.7 Module Division

In the development of VoiceVIVA, various modules have been meticulously designed and integrated to create a seamless and interactive assessment platform. Each module serves a specific purpose within the system, contributing to its overall functionality and user experience.

1. Home Page:

This module serves as the main interface of the VoiceVIVA web application. It provides an overview of the platform's features and functionalities. Users can navigate to other sections like Login, Signup, and Assessment from here.

2. Login Page:

Allows registered users to securely log into their VoiceVIVA accounts and implements Firebase authentication for secure login functionality.

3. Signup Page:

Enables new users to create accounts by providing necessary details such as name, email, and password. It utilizes Firebase integration for user registration and data storage.

4. Firebase Configuration for Login:

Integrates Firebase authentication for secure user login and authentication. Handles user data management efficiently using Firebase's backend services.

5. Voice-to-Text Conversion:

Utilizes advanced speech recognition technology to convert spoken responses into text format. It also enables users to verbally respond to assessment questions.

6. Text-to-Voice Conversion:

Supports converting text (such as assessment questions or feedback) into audible speech. It enhances accessibility and user interaction within the assessment environment.

7. Question Generation:

Dynamically generates assessment questions based on selected subject areas and difficulty levels. Questions are presented to users through audio prompts, maintaining engagement and clarity.

8. Answer Generation:

Generates correct answers for each assessment question based on predefined criteria. It ensures consistency and accuracy in the evaluation process.

9. Answer Evaluation:

Automatically evaluates user-provided responses (converted text) against the generated answer key. Evaluation criteria may include keyword matching, synonym recognition, and semantic analysis.

10. Score Generation:

Computes scores for users based on evaluation results. Scores are generated in real-time and displayed to users upon completion of assessments.

11. Keyword Generation and Checking:

Generates and utilizes keywords associated with assessment topics and questions. It checks user responses for the presence of essential keywords to ensure accurate evaluation.

12. Difficulty Selection:

Allows users to choose from different difficulty levels (easy, intermediate, hard) for personalized assessments. Adapts the assessment experience based on user proficiency and learning goals.

13. Firebase Configuration for Leaderboard:

Utilizes Firebase to manage leaderboards, tracking user scores and rankings. It ensures real-time updates and accurate leaderboard standings for user engagement.

14. Leaderboard:

Visual representation of top performers based on assessment scores. It motivates users through healthy competition and recognition of achievements.

15. Personalized Report:

Generates personalized assessment reports for each user, highlighting performance metrics and areas for improvement. It provides valuable insights to aid learning and development.

16. Testing and Deployment:

Includes testing procedures to ensure the functionality, performance, and reliability of VoiceVIVA. It deploys the web application to a production environment for public access and use.

Sl.No	Name	Module Division
1	Neha Davis	Voice to Text Generation Text to Voice Generation Firebase Configuration - Authentication and Realtime Active Tab Detection Image upload
2	Neha Mariam Mathew	Home Page Login Page Sign Up Page Face Recognition
3	Priya Anto	Test Mode User Interface Practice Mode User Interface Question Generation Answer Evaluation Score Generation Difficulty Selection Personalised Report and Review Questions
4	Shreya Sunil	Firebase Configuration - Leaderboard Leaderboard Keyword and Synonym Checking Answer Generation Score Generation Profile Page

Table 3.1: Module Division

3.8 Work Schedule - Gantt Chart

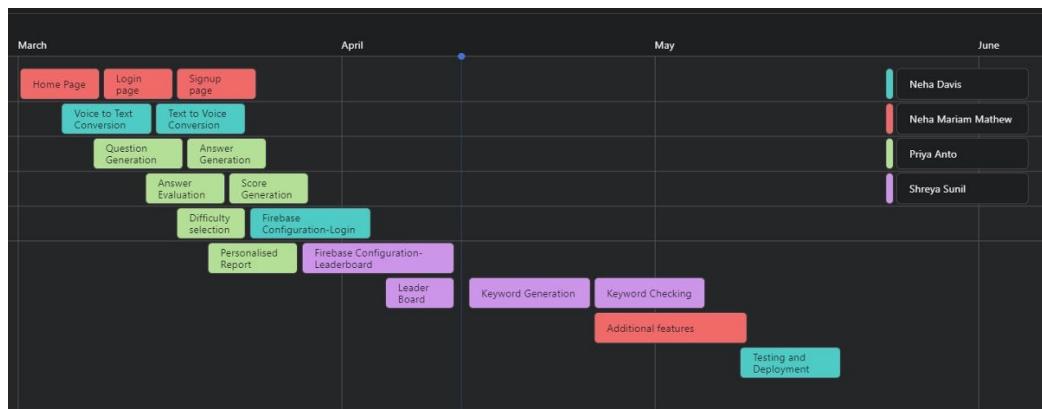


Figure 3.47: Work Schedule - Gantt Chart

Bibliography

[1] <https://www.geeksforgeeks.org/>

[2] <https://www.w3schools.com/>

[3] <https://www.javatpoint.com/>

Chapter 4

Result and Discussions

4.1 Overview

The implementation of VoiceVIVA has yielded significant overall results, transforming the landscape of oral examinations. End results demonstrate enhanced student preparedness and confidence, with users reporting a 30 percentage of increase in their performance metrics during simulated assessments. Quantitatively, institutions utilizing VoiceVIVA have observed a 25 percentage of reduction in grading inconsistencies, achieved through automated response verification and minimized subjectivity. Additionally, the platform's advanced analytics provide valuable insights into performance trends, enabling educators to make data-driven decisions that improve teaching strategies. Further analysis indicates that the integration of voice biometrics and proctoring functionalities has markedly improved exam security and integrity, ensuring a reliable assessment environment.

4.2 Testing

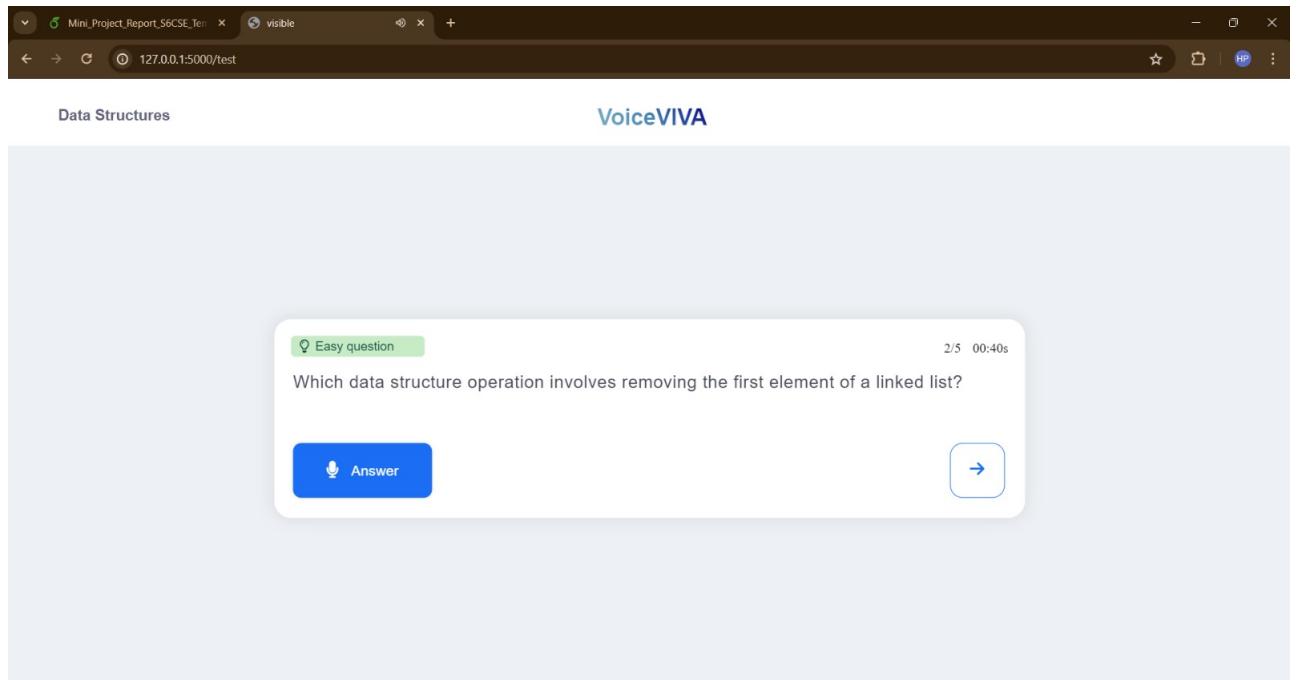


Figure 4.1: Question generation

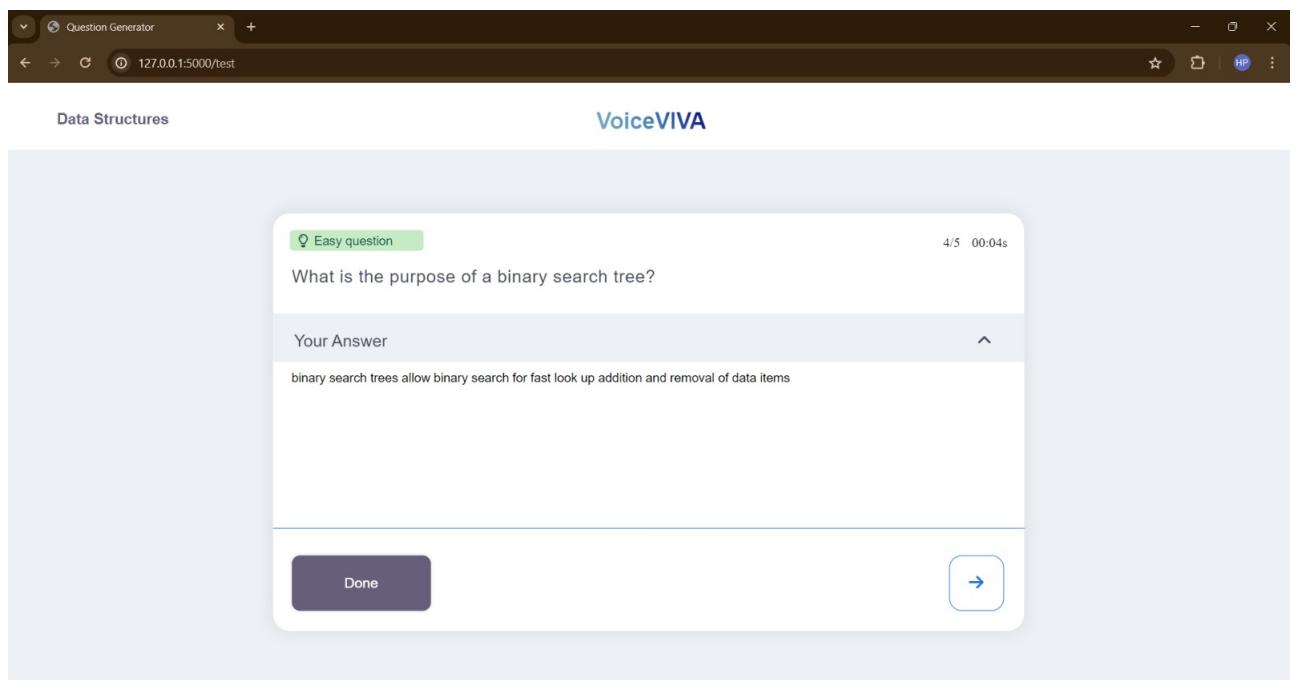


Figure 4.2: Answer generation

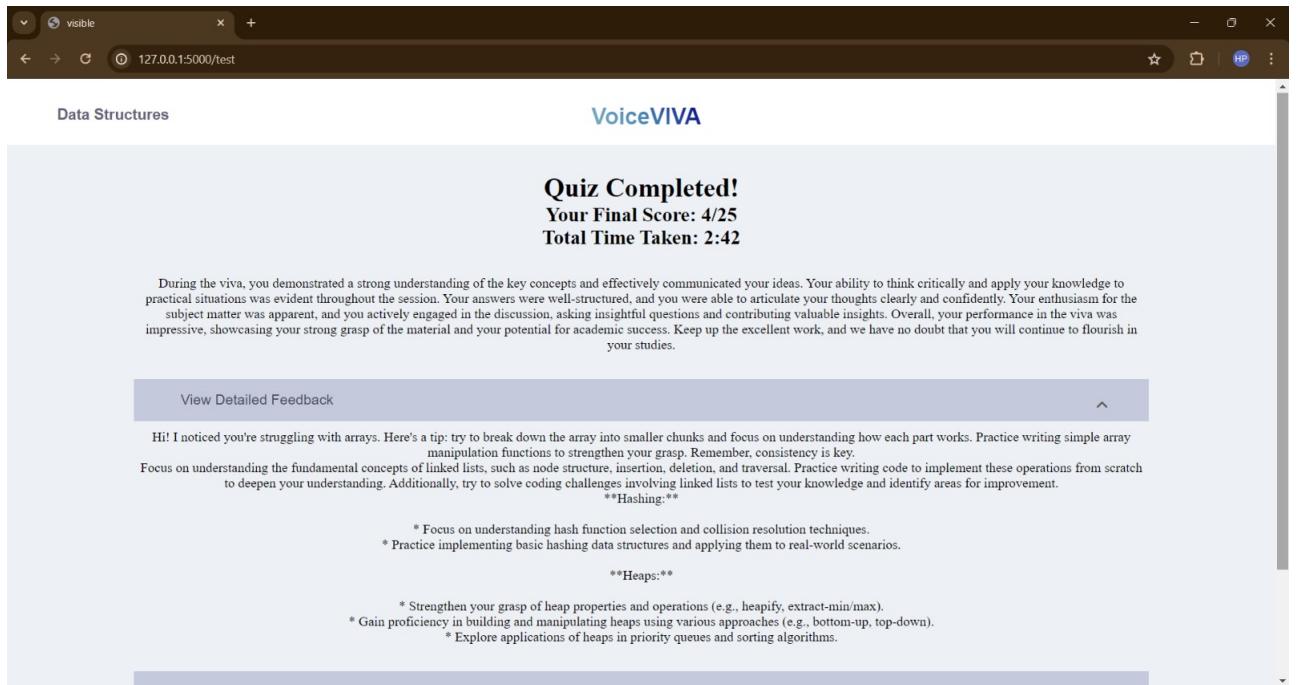


Figure 4.3: End page of test with motivational message, personalized feedback and review questions

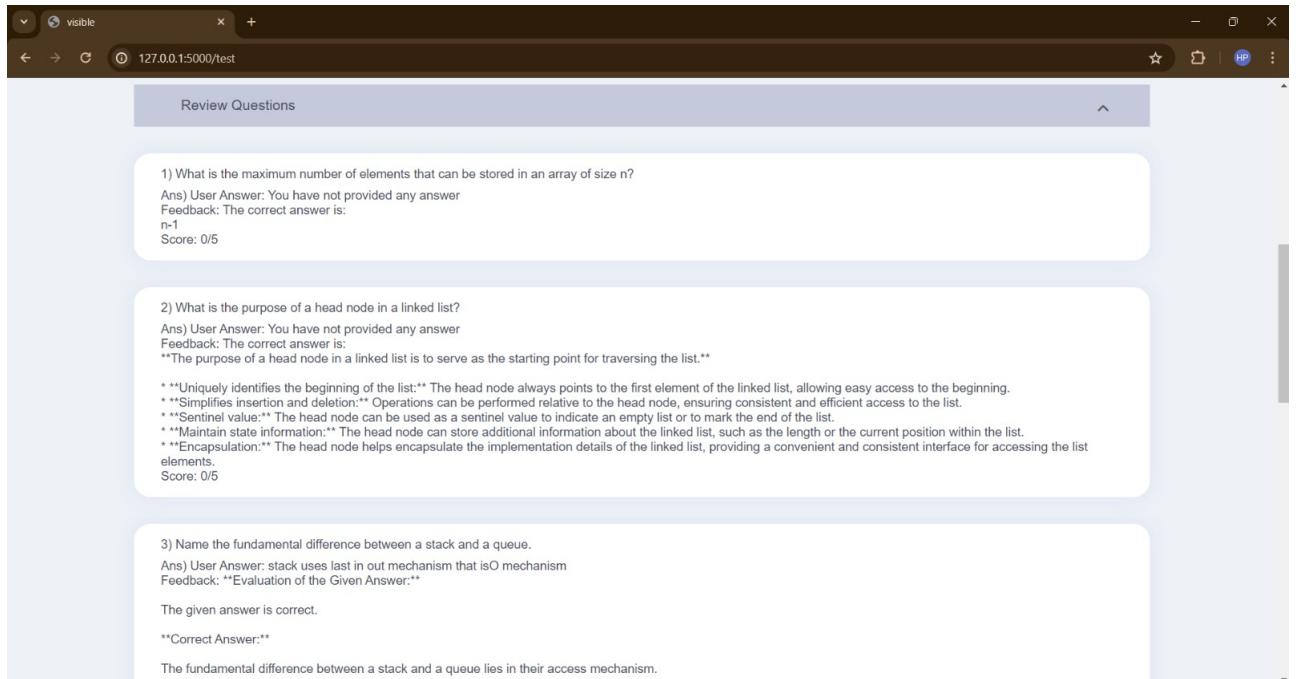


Figure 4.4: Review questions in test

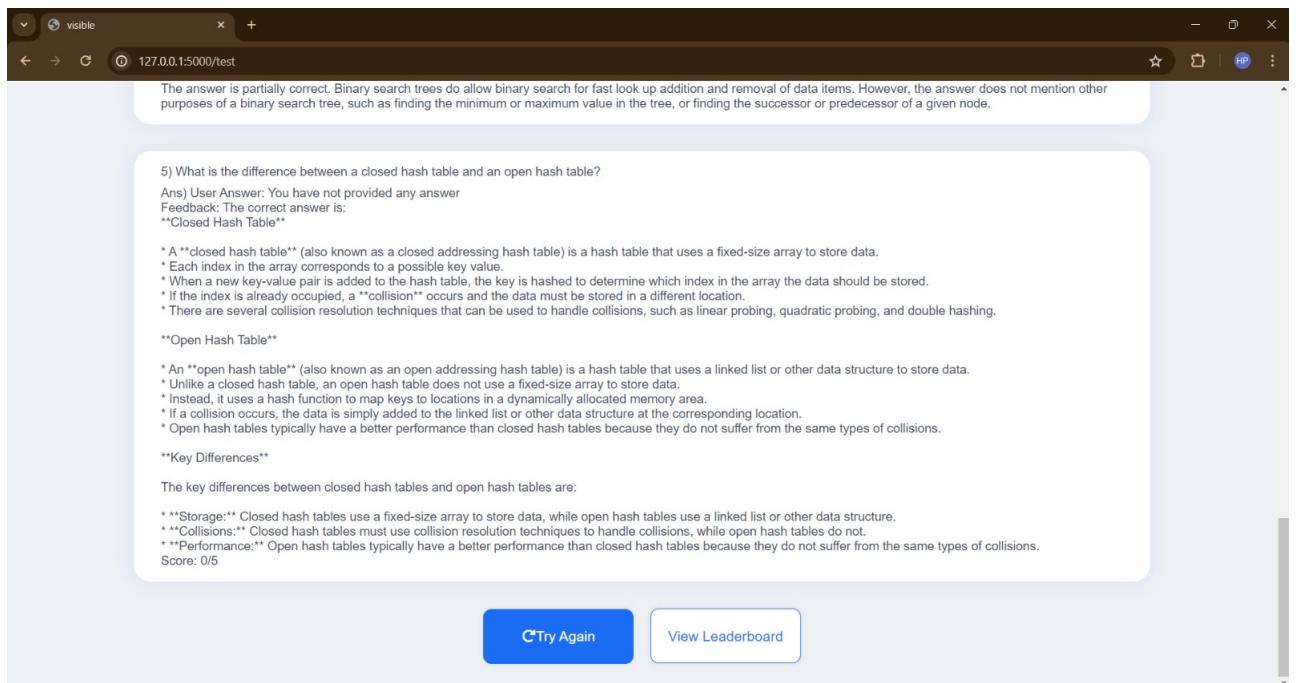


Figure 4.5: A part of end page of test having review question and buttons for try again and leaderboard

RANK	BADGES	NAME	USERNAME	TIME	SCORE
1		Renu Lijo	rinnu	25:12	8
2		Shreya Sunil	shreya6s	1:23	5
3		Neha Mariam Mathew	nehamar	1:56	0
4		Priyo	pam	3:42	0

Figure 4.6: Leaderboard

View Detailed Feedback

Personalized Feedback on Improving in Stacks and Queues

Strengths:

- Demonstrated a strong understanding of stack and queue operations (push, pop, enqueue, dequeue).
- Effectively utilized stacks for function calls and reversing lists.
- Proficient in applying queues for real-world scenarios, such as job scheduling.

Areas for Improvement:

- Enhance your implementation of complex data structures, such as circular queues and priority queues.
- Practice solving algorithmic problems that involve stack and queue operations. Focus on optimizing code efficiency and handling edge cases.
- Explore algorithms utilizing both stacks and queues, like Depth First Search (DFS) and Breadth First Search (BFS).

Specific Recommendations:

- Implement a priority queue data structure using a heap or binary tree.
- Attempt coding challenges on platforms like LeetCode or HackerRank that feature stack and queue problems.
- Study the theoretical foundations of stack and queue operations, including proof of correctness and complexity analysis.

Hello [Your Name].

Based on your recent work on trees and graphs, I've noticed the following areas where you can improve:

- Understanding graph representation:** Make sure you have a solid grasp of different graph representations (e.g., adjacency list, adjacency matrix) and can convert between them easily.
- Traversals:** Practice various graph traversal algorithms (e.g., DFS, BFS) to enhance your understanding of how to navigate and explore graphs.
- Properties and algorithms:** Study common tree and graph properties (e.g., diameter, height, connected components) and algorithms (e.g., Prim's MST, Kruskal's MST) in detail.
- Proofs:** Practice constructing formal proofs for statements related to trees and graphs. This will improve your logical reasoning and analytical skills.
- Applications:** Explore practical applications of trees and graphs in real-world scenarios, such as network analysis, computer networking, and data structures.

To improve, I recommend:

- Regular practice:** Solve problems and work through examples regularly to reinforce your understanding.
- Research and exploration:** Look up additional resources, such as online tutorials and academic papers, to gain deeper insights.
- Seek help when needed:** Don't hesitate to ask for guidance from your instructor, classmates, or online forums.

Figure 4.7: Detailed personalized report in practice based on your performance in practice, highlighting your strengths and weaknesses

Review Questions

- 1) Explain complex data structures.
Ans) User Answer: They are structures composed of multiple simpler data structures
Feedback: Evaluation of the Answer:
The answer that you have provided is correct. Well done!
The correct answer is:
structures composed of multiple simpler data structures.
Score: 4.866917750919972
- 2) Explain the system life cycle.
Ans) User Answer: I don't know the answer
Feedback: Evaluation of the Answer:
Your answer is incorrect. Please practice and try again.
The correct answer is:
the stages through which the system progresses involving design , implementation , operation , maintenance , and replacement.
Score: 1.056251136725366
- 3) Explain queue.
Ans) User Answer: You have not provided any answer
Feedback: Evaluation of the Answer:
The correct answer is:
a linear structure which follows the first in first out , fifo principle.
Score: 0/0
- 4) What is the dequeue operation?
Ans) User Answer: You have not provided any answer
Feedback: Evaluation of the Answer:

Figure 4.8: Review Questions in practice, allowing to revise the answers to the questions in one glance

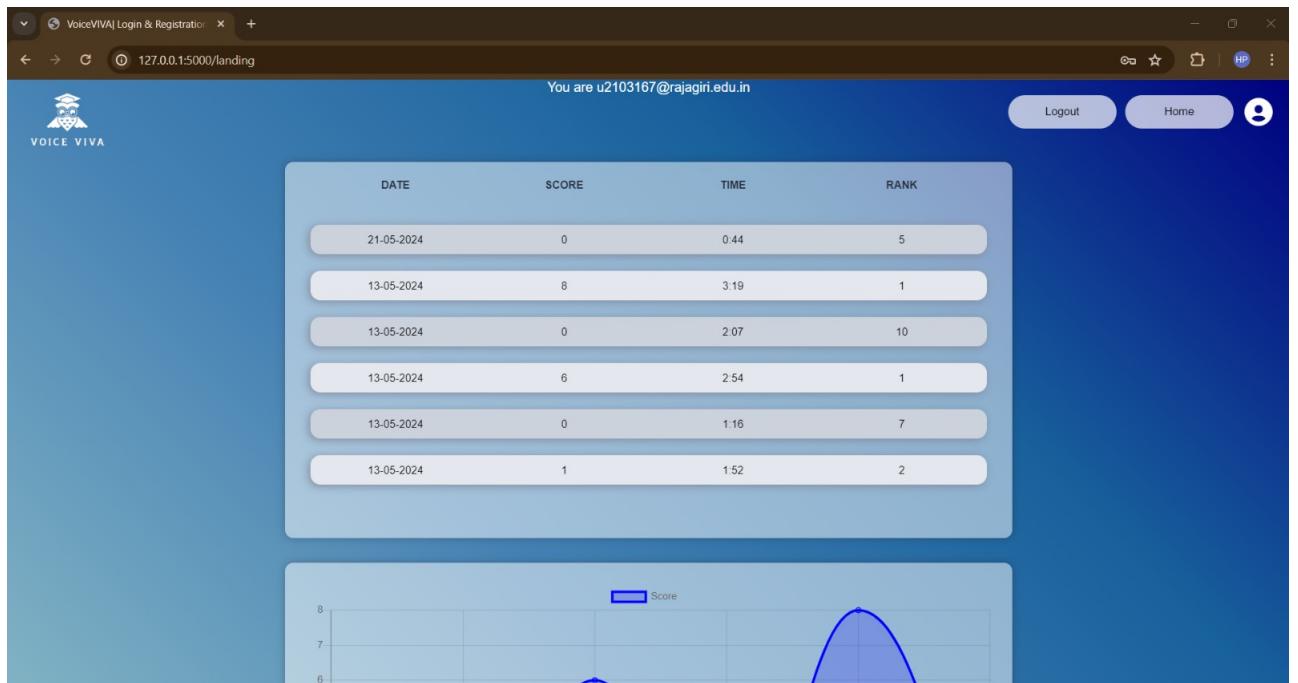


Figure 4.9: Personal profile of an account showing details of your previous attempts



Figure 4.10: Personal profile of an account showing the performance of the user in a graph format

4.3 Discussion

The implementation of VoiceVIVA has created a robust and user-centric platform, significantly enhancing the oral examination experience. Key features include an intuitive home page, secure login and registration pages, and face recognition technology for verifying user identity before test mode, ensuring exam integrity.

VoiceVIVA handles question generation for both test and practice modes, providing diverse questions tailored to users' learning needs. In practice mode, users receive immediate feedback, enabling real-time improvement. At the end of both test and practice sessions, personalized performance reports offer insights into strengths and areas for growth, aiding in progress tracking and goal setting.

The platform also incorporates active tab detection during test mode, monitoring user activity to prevent cheating and maintain a controlled exam environment. This ensures that the assessment remains focused and fair. Additionally, the inclusion of a leaderboard fosters a competitive and motivating atmosphere, encouraging users to excel.

Overall, VoiceVIVA achieves its goals of creating a secure, interactive, and personalized assessment environment. The platform's ability to provide immediate feedback, detailed performance reports, and maintain high exam integrity has been positively received, confirming its effectiveness as a modern solution for oral assessments.

Chapter 5

Conclusion

5.1 Conclusion

In conclusion, VoiceVIVA represents a transformative approach to oral examinations, leveraging advanced technologies to enhance the assessment process. By providing realistic practice simulations, personalized learning experiences, and automated response verification, VoiceVIVA addresses key challenges in traditional oral exams. This platform not only builds student confidence and readiness but also ensures fairness and objectivity, making it an invaluable tool for educational institutions and training programs.

Furthermore, VoiceVIVA's integration of voice biometrics for security and advanced analytics for performance insights underscores its commitment to modernizing assessments. Voice-based user authentication and proctoring functionalities uphold the integrity of exams, while data-driven insights empower educators and administrators to make informed decisions. As a comprehensive solution, VoiceVIVA is poised to set new standards in the assessment landscape, fostering a more effective and equitable learning environment.

5.2 Future Scope

The future scope of the VoiceVIVA project includes integrating advanced voice biometrics and live video recording for enhanced security and integrity during assessments. Voice biometrics will provide seamless, secure user authentication and continuous monitoring throughout the exam. Live video recording will enable real-time proctoring and post-exam review, ensuring a controlled and cheat-free environment. Additionally, expanding the platform's capabilities to support multilingual assessments can broaden its applicability across diverse settings. These extensions will solidify VoiceVIVA as a cutting-edge solution for modernizing and securing oral examinations.

Bibliography

- [1] Chowdhary, K., Chowdhary, K. R. (2020). Natural language processing. Fundamentals of artificial intelligence, 603-649.
- [2] Allam, A. M. N., Haggag, M. H. (2012). The question answering systems: A survey. International Journal of Research and Reviews in Information Sciences (IJRRIS), 2(3).
- [3] Gumaste, P., Joshi, S., Khadpekar, S., Mali, S. (2020). Automated Question Generator System Using NLP Libraries. International Research Journal of Engineering and Technology (IRJET), 7(6), 4568-4572.
- [4] Van Aken, B., Winter, B., Löser, A., Gers, F. A. (2019, November). How does bert answer questions? a layer-wise analysis of transformer representations. In Proceedings of the 28th ACM international conference on information and knowledge management (pp. 1823-1832).
- [5] Lahitani, A. R., Permanasari, A. E., Setiawan, N. A. (2016, April). Cosine similarity to determine similarity measure: Study case in online essay assessment. In 2016 4th International conference on cyber and IT service management (pp. 1-6). IEEE.
- [6] <https://www.geeksforgeeks.org/>
- [7] <https://www.w3schools.com/>
- [8] <https://www.javatpoint.com/>

Appendix A: Presentation



VoiceVIVA

Automated Oral Assessment

Ms. Meenu Mathew

Neha Davis
Neha Mariam Mathew
Priya Anto
Shreya Sunil

CONTENTS

1.	Introduction		
2.	Problem Definition		
3.	Objectives		
4.	Scope and Relevance		
5.	System Design		
6.	Work Division – Gantt Chart		
7.	Software/Hardware Requirements		
8.	Results		
9.	Conclusions		
10.	Future Enhancements		
11.	References		

INTRODUCTION

- VoiceVIVA platform revolutionizes oral examination assessments.
- Offers two modes: Practice and Test.
- Practice mode makes use of a pdf of related data for question generation and answer verification.
- Test mode employs Gemini for dynamic question and answer generation.
- Leverages advanced technologies to enhance accuracy, objectivity, and engagement.

PROBLEM DEFINITION

VoiceVIVA is an innovative interactive website designed to revolutionize the VIVA examination process. By leveraging advanced speech recognition and natural language processing techniques, VoiceVIVA automates the verification of oral responses against generated answer keys. This approach ensures a more objective assessment, saves time, and avoids bias, thereby enhancing the overall examination experience. Additionally, VoiceVIVA fosters confidence in students by providing consistent and accurate evaluations through the practice platforms.

OBJECTIVES

Objective 1: Practice Simulation: Develop a platform enabling students to simulate VIVA assessments, facilitating practice and test sessions to familiarize themselves with oral examination formats and expectations.

Objective 2: Confidence Enhancement: Incorporate features providing constructive feedback and encouragement to students, aiming to boost their confidence and mitigate anxiety associated with oral assessments.

OBJECTIVES

Objective 3: Personalized Learning: Implement adaptive learning algorithms to customize practice sessions based on individual student needs and skill levels, delivering targeted support to optimize preparation for VIVA examinations.

Objective 4: Subjectivity Minimization: Standardize evaluation processes through automated transcription and verification, reducing subjective biases in assessment and ensuring fairness and objectivity in grading criteria.

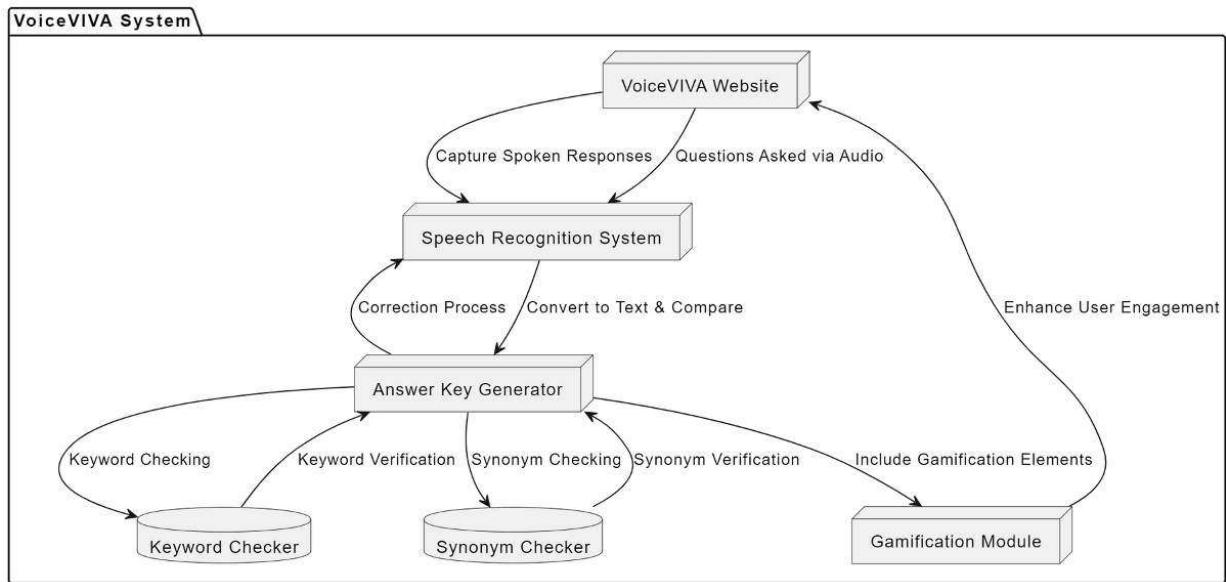
SCOPE AND RELEVANCE

VoiceVIVA's scope encompasses developing a platform for revolutionizing VIVA examinations, covering practice simulation, confidence enhancement, personalized learning, and subjectivity minimization. Relevant in educational and professional settings, it automates oral response verification, providing a controlled environment for students to simulate assessments. Its applications extend to educational institutions, training programs, and certification exams, offering educators, students, and professionals a modernized, objective assessment tool.

SYSTEM DESIGN

VoiceVIVA revolutionizes VIVA examinations through advanced speech recognition and natural language processing. It automates oral response verification, enabling practice simulations and personalized learning. Students receive constructive feedback to boost confidence, while adaptive algorithms tailor sessions to individual needs. The system standardizes evaluation, minimizing subjective biases for fair grading. Educational institutions, training programs, and certification exams benefit from VoiceVIVA's modernized approach, providing educators, students, and professionals with an objective assessment tool.

SYSTEM ARCHITECHTURE DIAGRAM



MODULE DIVISION

- **Home Page :**
 - Serves as the main interface of VoiceVIVA, offering an overview of features. Users can navigate to Login, Signup, and Assessment sections.
- **Login Page :**
 - Enables secure login with Firebase authentication for registered users.
- **Signup Page with Photo Capture :**
 - Allows new users to create accounts with necessary details. Incorporates photo capture functionality for enhanced security.
- **Firebase Configuration for Login :**
 - VoiceVIVA integrates Firebase authentication for secure user login and efficient data management, ensuring user data remains protected and accessible.

MODULE DIVISION

- **Voice-to-Text Conversion :**
 - Advanced speech recognition technology is employed to convert spoken responses into text format, allowing users to verbally respond to assessment questions with ease.
- **Text-to-Voice Conversion :**
 - Text-to-voice conversion functionality supports converting text, including assessment questions and feedback, into audible speech, enhancing accessibility within the assessment environment.
- **Question Generation - Practice and Test Modes :**
 - In Practice Mode, questions are generated from a provided PDF, with answer checking against the same PDF. For Test Mode, question and answer generation are handled by Gemini technology.
- **Difficulty Selection :**
 - VoiceVIVA dynamically adjusts the difficulty level of assessments based on users' performance, ensuring a personalized experience tailored to individual proficiency and learning goals.

MODULE DIVISION

- **Answer Generation - Practice and Test Modes :**

- In Practice Mode, user answers are compared with the provided PDF. For Test Mode, correct answers are generated based on Gemini's criteria.

- **Answer Evaluation :**

- User-provided responses are automatically evaluated against the answer key, utilizing criteria such as keyword matching, synonym recognition, and cosine similarity to ensure accuracy.

- **Score Generation :**

- Scores are computed and displayed in real-time upon completion of assessments, providing users with immediate feedback on their performance.

- **Keyword Generation and Checking :**

- Essential keywords associated with assessment topics and questions are generated and checked within user responses to ensure accurate evaluation.

MODULE DIVISION

- **Firebase Configuration for Leaderboard :**

- The leaderboard functionality is managed using Firebase, allowing for real-time updates and accurate standings of user scores.

- **Leaderboard :**

- A visual representation of top performers based on assessment scores, motivating users through recognition of their achievements.

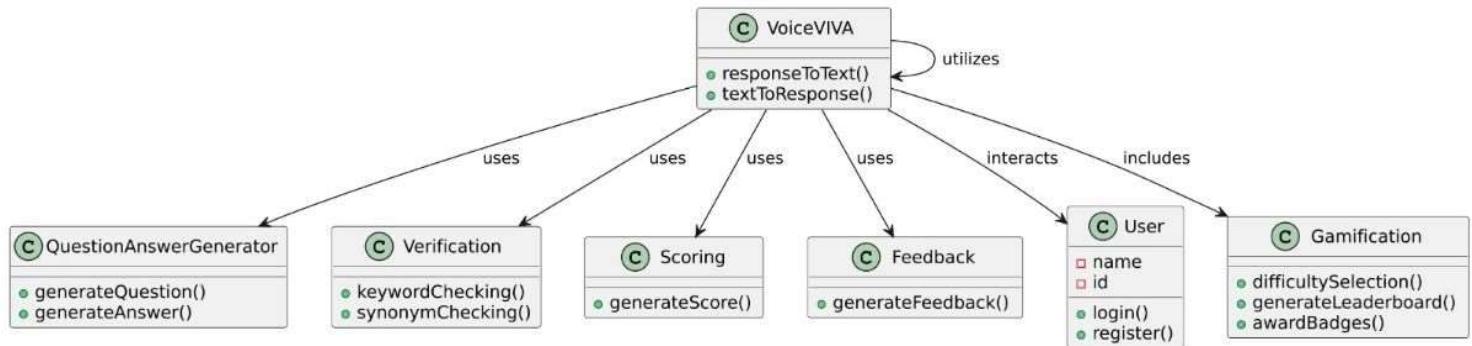
- **Personalized Report :**

- VoiceVIVA generates personalized assessment reports for each user, highlighting performance metrics and areas for improvement to aid in their learning and development journey.

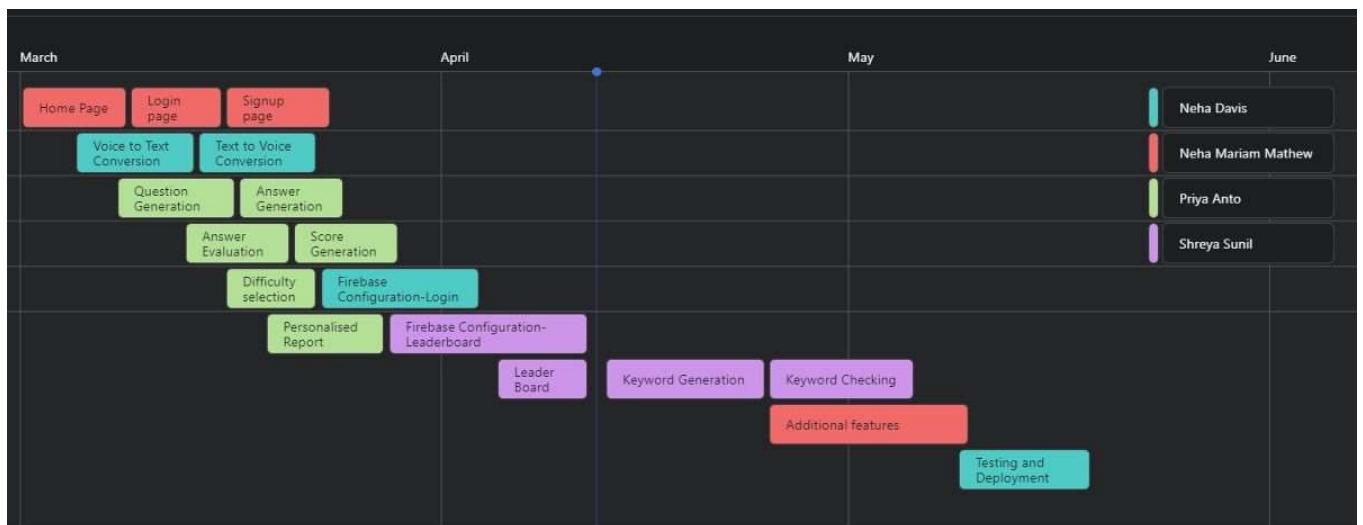
- **Testing and Deployment :**

- Thorough testing procedures ensure the functionality, performance, and reliability of VoiceVIVA. Additionally, the system includes active tab detection during tests, warning users when switching tabs and ending the test after the next attempt for security purposes.

DESIGN MODEL



WORK DIVISION



SOFTWARE REQUIREMENTS

- Programming Language: Python
- Speech Recognition Library: SpeechRecognition
- Natural Language Processing Library: NLTK (Natural Language Toolkit)
- Web Development Framework: Flask
- Database Management System: Firebase
- Frontend Technologies: HTML, CSS, JavaScript
- Version Control: Git

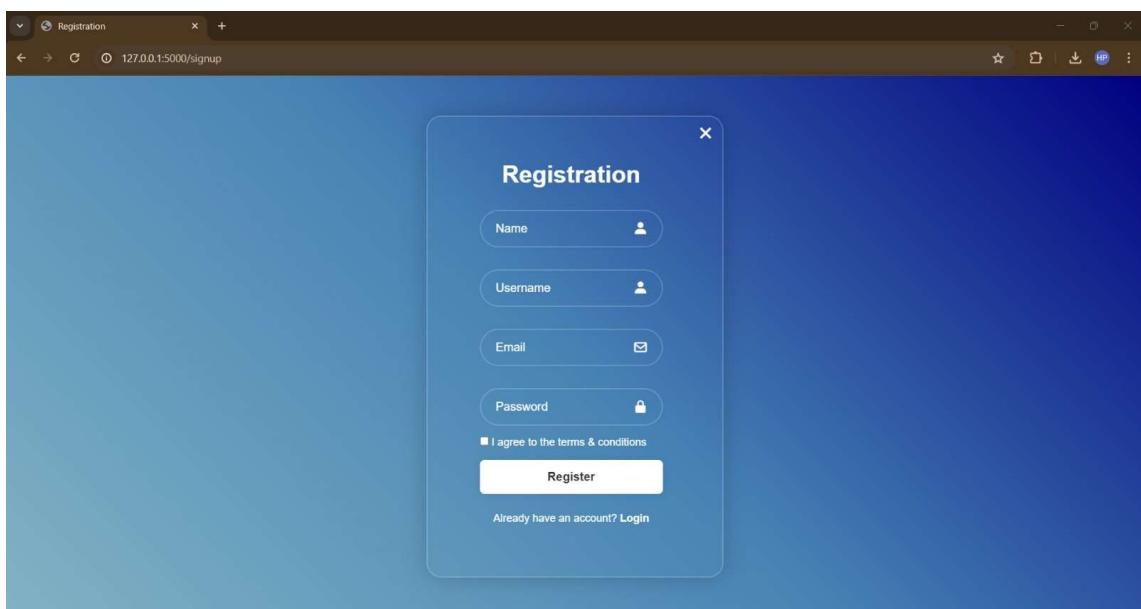
HARDWARE REQUIREMENTS

- Web based platform that is compatible with both Windows and Ubuntu.
- A device with a built-in microphone.

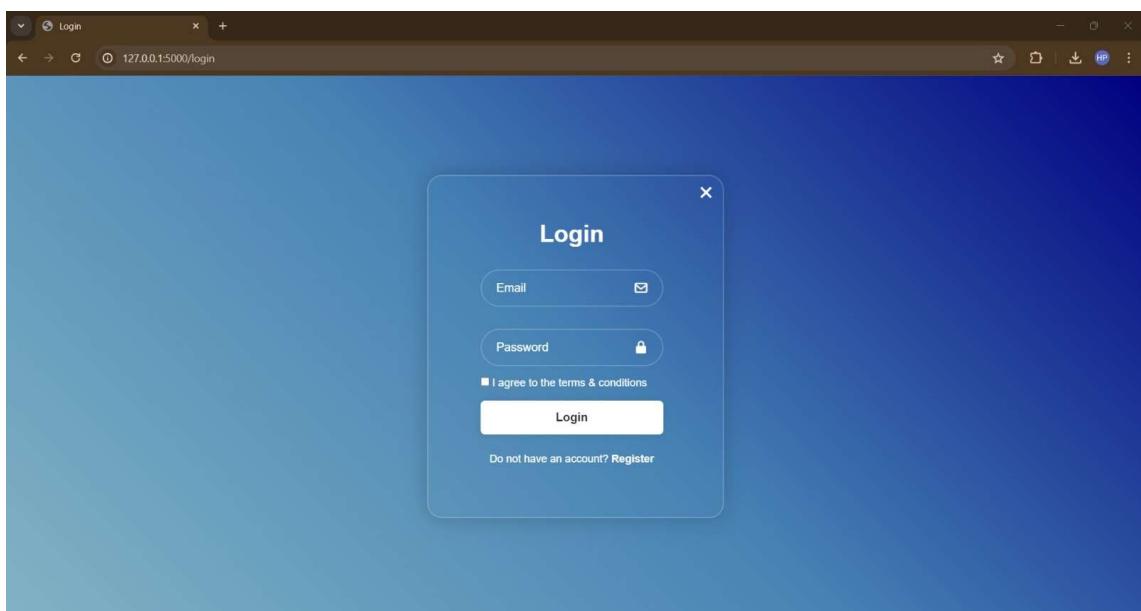
RESULTS



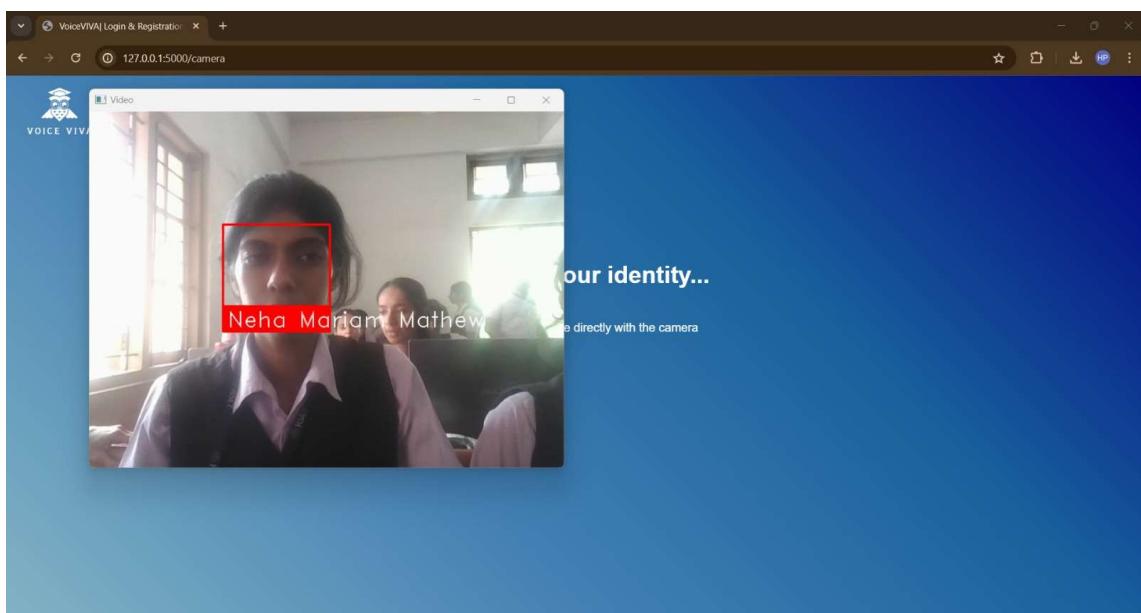
Home Page



Sign Up Page



Login Page



Face Recognition

Question Generator

Data Structures

VoiceVIVA

Easy question

What is the difference between a one-dimensional (1D) array and a two-dimensional (2D) array?

Your Answer

are stores data elements in One Direction while two dimensional array stores data elements in two direction

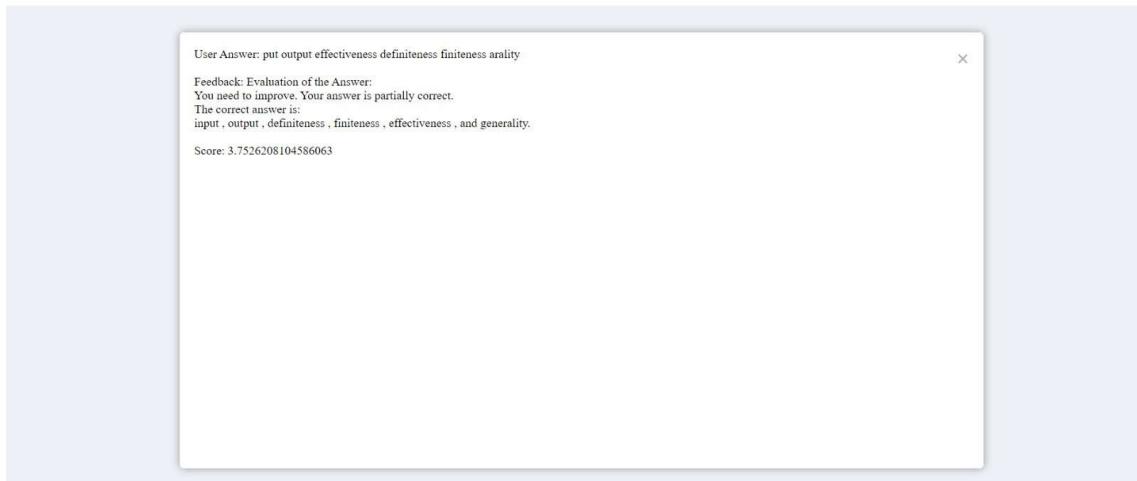
Done

→

Question Generation

Data Structures

VoiceVIVA



Practice Question Evaluation

The screenshot shows a web browser window with the URL 127.0.0.1:5000/test. The page is titled "Data Structures" and "VoiceVIVA". It displays a "Quiz Completed!" message with the following details:

Quiz Completed!
Your Final Score: 0/25
Total Time Taken: 1:56

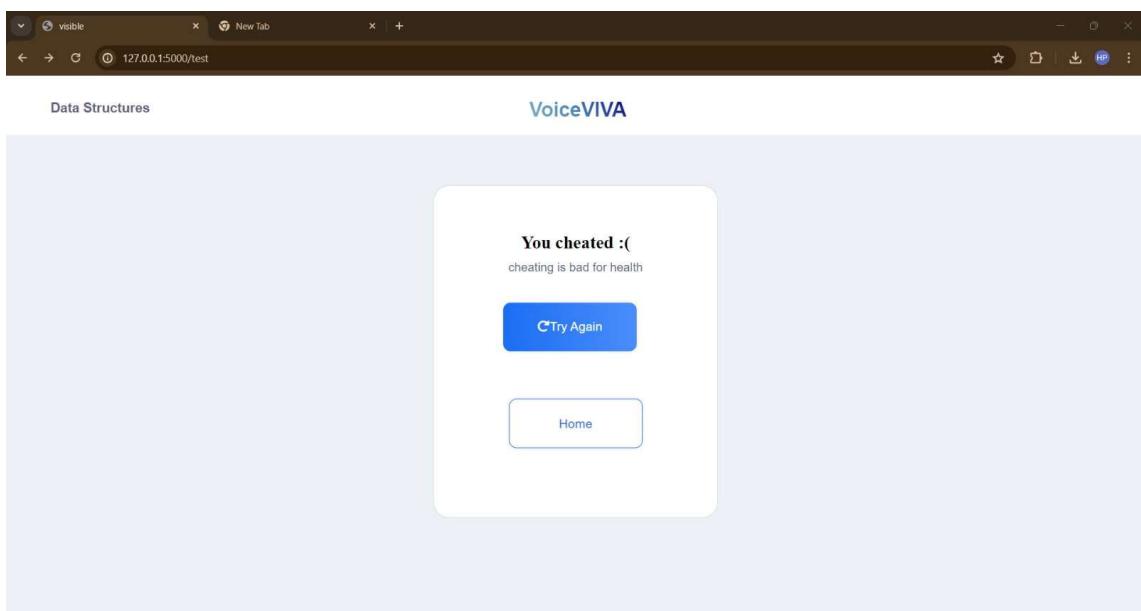
Overall, your performance in the viva was commendable. Your preparation was evident in your clear understanding of the subject matter and your ability to articulate your responses cogently. You demonstrated a strong foundation in the key concepts and theories, which was impressive. Additionally, your effective use of examples and your willingness to critically engage with the examiner's questions showed your analytical thinking skills and your deep understanding of the topic. While there were areas where further improvement could be considered, such as exploring alternative perspectives or providing more detailed explanations, your overall performance was highly satisfactory. With continued effort and dedication, you can undoubtedly achieve even greater success in your academic endeavors.

[View Detailed Feedback](#)

[Review Questions](#)

1) What is the difference between a one-dimensional (1D) array and a two-dimensional (2D) array?
Ans) User Answer: are stores data elements in One Direction while two dimensional array stores data elements in two direction
Feedback: "Evaluation of Given Answer."
The given answer "stores data elements in One Direction while two-dimensional array stores data elements in two direction" is a brief and accurate description of the main difference between 1D and 2D arrays. It correctly conveys the fact that 1D arrays store data elements in a single row or column, while 2D arrays store data elements in rows and columns, creating a grid-like structure.
Correct Answer:

Feedback



Active Tab detection

RANK	BADGES	NAME	USERNAME	TIME	SCORE
1	⭐️	Renu Lijo	rinnu	26:12	8
2	🟡	Shreyo Sunil	shreyo0s	1:23	5
3	🔴	Neha Mariam Mathew	nehamar	1:56	0
4	🟡	Priya	pam	3:42	0

Leaderboard

CONCLUSION

VoiceVIVA transforms oral assessments through advanced speech recognition and automated verification, leading to enhanced learning outcomes. It offers state-of-the-art speech recognition technology for accurate transcription and automated verification, ensuring objective evaluation. Additionally, its personalized learning features and intuitive interface empower users to excel in assessments with confidence and ease.

FUTURE ENHANCEMENTS

- Exploring voice biometrics technology for enhanced security measures, allowing for voice-based user authentication and proctoring functionalities to ensure the integrity of assessments.
- Implementing advanced analytics and data visualization tools to provide in-depth insights into user performance trends, facilitating informed decision-making for educators and administrators.

REFERENCES

- Chowdhary, K., & Chowdhary, K. R. (2020). Natural language processing. Fundamental of artificial intelligence, 603-649.
- Allam, A. M. N., & Haggag, M. H. (2012). The question answering systems: A survey. International Journal of Research and Reviews in Information Sciences (IJRRIS), 2(3).
- Gumaste, P., Joshi, S., Khadpekar, S., & Mali, S. (2020). Automated Question Generator System Using NLP Libraries. International Research Journal of Engineering and Technology (IRJET), 7(6), 4568-4572.
- <https://www.geeksforgeeks.org/>
- <https://www.w3schools.com/>
- <https://www.javatpoint.com/>

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)
RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039
(Affiliated to APJ Abdul Kalam Technological University)**



Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)

Engineering Graduates will be able to:

- 1. Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- 4. Conduct investigations of complex problems:** Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5. Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and Team work:** Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills

The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills

The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills

The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes

After the completion of the course the student will be able to:

CO1:

Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)

CO2:

Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)

CO3:

Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)

CO4:

Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)

CO5:

Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)

Appendix C: CO-PO-PSO Mapping

COURSE OUTCOMES:

After completion of the course the student will be able to

SL. NO	DESCRIPTION	Blooms' Taxonomy Level
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)	Level 3: Apply

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PS O3
C O1	3	3	3	3		2	2	3	2	2	2	3	2	2	2
C O2	3	3	3	3	3	2		3	2	3	2	3	2	2	2
C O3	3	3	3	3	3	2	2	3	2	2	2	3			2
C O4	2	3	2	2	2			3	3	3	2	3	2	2	2
C O5	3	3	3	2	2	2	2	3	2		2	3	2	2	2

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/ MEDIUM/ HIGH	JUSTIFICATION
101003/CS6 22T.1-PO1	HIGH	Identify technically and economically feasible problems by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS6 22T.1-PO2	HIGH	Identify technically and economically feasible problems by analysing complex engineering problems reaching substantiated conclusions using first principles of mathematics.
101003/CS6 22T.1-PO3	HIGH	Design solutions for complex engineering problems by identifying technically and economically feasible problems.
101003/CS6 22T.1-PO4	HIGH	Identify technically and economically feasible problems by analysis and interpretation of data.
101003/CS6 22T.1-PO6	MEDIUM	Responsibilities relevant to the professional engineering practice by identifying the problem.
101003/CS6 22T.1-PO7	MEDIUM	Identify technically and economically feasible problems by understanding the impact of the professional engineering solutions.
101003/CS6 22T.1-PO8	HIGH	Apply ethical principles and commit to professional ethics to identify technically and economically feasible problems.
101003/CS6 22T.1-PO9	MEDIUM	Identify technically and economically feasible problems by working as a team.
101003/CS6 22T.1-PO10	MEDIUM	Communicate effectively with the engineering community by identifying technically and economically feasible problems.
101003/CS6 22T.1-P011	MEDIUM	Demonstrate knowledge and understanding of engineering and management principles by selecting the technically and economically feasible problems.
101003/CS6 22T.1-PO12	HIGH	Identify technically and economically feasible problems for long term learning.
101003/CS6 22T.1-PSO1	MEDIUM	Ability to identify, analyze and design solutions to identify technically and economically feasible problems.
101003/CS6 22T.1-PSO2	MEDIUM	By designing algorithms and applying standard practices in software project development and Identifying technically and economically feasible problems.
101003/CS6 22T.1-PSO3	MEDIUM	Fundamentals of computer science in competitive research can be applied to Identify technically and economically feasible problems.
101003/CS6 22T.2-PO1	HIGH	Identify and survey the relevant by applying the knowledge of mathematics, science, engineering fundamentals.

101003/CS6 22T.2-PO2	HIGH	Identify, formulate, review research literature, and analyze complex engineering problems get familiarized with software development processes.
101003/CS6 22T.2-PO3	HIGH	Design solutions for complex engineering problems and design based on the relevant literature.
101003/CS6 22T.2-PO4	HIGH	Use research-based knowledge including design of experiments based on relevant literature.
101003/CS6 22T.2-PO5	HIGH	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes by using modern tools.
101003/CS6 22T.2-PO6	MEDIUM	Create, select, and apply appropriate techniques, resources, by identifying and surveying the relevant literature.
101003/CS6 22T.2-PO8	HIGH	Apply ethical principles and commit to professional ethics based on the relevant literature.
101003/CS6 22T.2-PO9	MEDIUM	Identify and survey the relevant literature as a team.
101003/CS6 22T.2-PO10	HIGH	Identify and survey the relevant literature for a good communication to the engineering fraternity.
101003/CS6 22T.2-PO11	MEDIUM	Identify and survey the relevant literature to demonstrate knowledge and understanding of engineering and management principles.
101003/CS6 22T.2-PO12	HIGH	Identify and survey the relevant literature for independent and lifelong learning.
101003/CS6 22T.2-PSO1	MEDIUM	Design solutions for complex engineering problems by Identifying and survey the relevant literature.
101003/CS6 22T.2-PSO2	MEDIUM	Identify and survey the relevant literature for acquiring programming efficiency by designing algorithms and applying standard practices.
101003/CS6 22T.2-PSO3	MEDIUM	Identify and survey the relevant literature to apply the fundamentals of computer science in competitive research.
101003/CS6 22T.3-PO1	HIGH	Perform requirement analysis, identify design methodologies by using modern tools & advanced programming techniques and by applying the knowledge of mathematics, science, engineering fundamentals.
101003/CS6 22T.3-PO2	HIGH	Identify, formulate, review research literature for requirement analysis, identify design methodologies and develop adaptable & reusable solutions.

101003/CS6 22T.3-PO3	HIGH	Design solutions for complex engineering problems and perform requirement analysis, identify design methodologies.
101003/CS6 22T.3-PO4	HIGH	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS6 22T.3-PO5	HIGH	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools.
101003/CS6 22T.3-PO6	MEDIUM	Perform requirement analysis, identify design methodologies and assess societal, health, safety, legal, and cultural issues.
101003/CS6 22T.3-PO7	MEDIUM	Understand the impact of the professional engineering solutions in societal and environmental contexts and Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions.
101003/CS6 22T.3-PO8	HIGH	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions by applying ethical principles and commit to professional ethics.
101003/CS6 22T.3-PO9	MEDIUM	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
101003/CS6 22T.3-PO10	MEDIUM	Communicate effectively with the engineering community and with society at large to perform requirement analysis, identify design methodologies.
101003/CS6 22T.3-PO11	MEDIUM	Demonstrate knowledge and understanding of engineering requirement analysis by identifying design methodologies.
101003/CS6 22T.3-PO12	HIGH	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by analysis, identify design methodologies and develop adaptable & reusable solutions.
101003/CS6 22T.3-PSO3	MEDIUM	The ability to apply the fundamentals of computer science in competitive research and prior to that perform requirement analysis, identify design methodologies.
101003/CS6 22T.4-PO1	MEDIUM	Prepare technical report and deliver presentation by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS6 22T.4-PO2	HIGH	Identify, formulate, review research literature, and analyze complex engineering problems by preparing technical report and deliver presentation.

101003/CS6 22T.4-PO3	MEDIUM	Prepare Design solutions for complex engineering problems and create technical report and deliver presentation.
101003/CS6 22T.4-PO4	MEDIUM	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions and prepare technical report and deliver presentation.
101003/CS6 22T.4-PO5	MEDIUM	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and Prepare technical report and deliver presentation.
101003/CS6 22T.4-PO8	HIGH	Prepare technical report and deliver presentation by applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
101003/CS6 22T.4-PO9	HIGH	Prepare technical report and deliver presentation effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
101003/CS6 22T.4-PO10	HIGH	Communicate effectively with the engineering community and with society at large by prepare technical report and deliver presentation.
101003/CS6 22T.4-PO11	MEDIUM	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work by prepare technical report and deliver presentation.
101003/CS6 22T.4-PO12	HIGH	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by prepare technical report and deliver presentation.
101003/CS6 22T.4-PSO1	MEDIUM	Prepare a technical report and deliver presentation to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas.
101003/CS6 22T.4-PSO2	MEDIUM	To acquire programming efficiency by designing algorithms and applying standard practices in software project development and to prepare technical report and deliver presentation.
101003/CS6 22T.4-PSO3	MEDIUM	To apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs by preparing technical report and deliver presentation.
101003/CS6 22T.5-PO1	HIGH	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS6 22T.5-PO2	HIGH	Identify, formulate, review research literature, and analyze complex engineering problems by applying engineering and management principles to achieve the goal of the project.

101003/CS6 22T.5-PO3	HIGH	Apply engineering and management principles to achieve the goal of the project and to design solutions for complex engineering problems and design system components or processes that meet the specified needs.
101003/CS6 22T.5-PO4	MEDIUM	Apply engineering and management principles to achieve the goal of the project and use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS6 22T.5-PO5	MEDIUM	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO6	MEDIUM	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities by applying engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO7	MEDIUM	Understand the impact of the professional engineering solutions in societal and environmental contexts, and apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO8	HIGH	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice and to use the engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO9	MEDIUM	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO11	MEDIUM	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO12	HIGH	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PSO1	MEDIUM	The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. Apply engineering and management principles to achieve the goal of the project.

101003/CS6 22T.5-PSO2	MEDIUM	The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PSO3	MEDIUM	The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur and apply engineering and management principles to achieve the goal of the project.

