# EXAM APP FOR VISUAL IMPAIRED PERSON

**A PROJECT REPORT**

***Submitted by***

**DHANUSH VARTHAN R.V(211420104322)**
**UDAYAKUMAR.G(211420104341)**
**PRASANNA KUMAR U. (211420104331)**

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

***of***

## BACHELOR OF ENGINEERING

**IN**

**COMPUTER SCIENCE AND ENGINEERING**


## PANIMALAR ENGINEERING COLLEGE

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**APRIL 2024**

# PANIMALAR ENGINEERING COLLEGE

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

## BONAFIDE CERTIFICATE

Certified that this project report EXAM APP FOR VISUAL IMPAIRED PERSON is the bonafide work of DHANUSH VARTHAN R.V(211420104322) UDAYAKUMAR.G(211420104341) PRASANNA KUMAR U. (211420104331) who carried out the project work under my supervision.

| | |
|---|---|
| **Signature of the HOD with date** | **Signature of the Supervisor with date** |
| **DR L.JABASHEELA M.E.,Ph.D.,**<br>**Professor and Head,** | **Dr. Vinmathi M S**<br>**Associate Professor** |
| Department of Computer Science and Engineering, | Department of Computer Science and Engineering, |
| Panimalar Engineering College, | Panimalar Engineering College, |
| Chennai - 123 | Chennai - 123 |

Submitted for the Project Viva – Voce examination held on _____

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

# DECLARATION BY THE STUDENT

We DHANUSH VARTHANR.V (211420104322) UDAYAKUMAR.G (211420104341) PRASANNA KUMAR U. (211420104331) hereby declare that this project report titled “EXAM APP FOR VISUAL IMPAIRED PERSON.”, under the guidance of Dr. Vinmathi M S .is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

UDAYAKUMAR.G(211420104341)

PRASANNA KUMAR U. (211420104331)

N DHANUSH VARTHAN R.V(211420104322)

# ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his benevolent words and fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking cof this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express my heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Dr. N.Pugazhendi** and **Dr. Vinmathi M S** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

UDAYAKUMAR.G(211420104341)

PRASANNA KUMAR U. (211420104331)

N DHANUSH VARTHAN R.V(211420104322)

# ABSTRACT

The growth of the Internet, and in particular the Web, is already influence the way science is taught and will undoubtedly do so to greater extent in the future. In areas of education, it offers a medium that has the potential to be more responsive to students. Web based Examination system could be used via Internet or intranet for managing student examination. In the future blind peoples also can-do online exam like a normal human if test can be taken using an Android Application, Here Questions and choice are through in Voice our project is delivered in real time. Marking the test is done automatically and instantaneously the faculty is comforted from these, time consuming duties. Different versions of the same question can be generated for different students. Tests can be taken anytime and anywhere. Blinds can login the exam using their application. The Answer will be getting from the user through voice and the results also delivered through voice. The marks are automatically collected, analyzed, and distributed for purposes like evaluation of teaching and learning process. In this world blind peoples can get the knowledge using various interfaces. Such as one common method of the blind people has been intended to use is by voice recognition. For example, in Android Phone there is a facility which allows a blind people to get interact with that App as a normal man. If they touch any button or the display option in the Phone has been designed to react as soon as by giving voice output. So, a visor people can use all the facility provided in that Android phone (they are able to access all the options like a normal man).

TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF ABBREVATION

| ABBREVATION | DESCRIPTION |
|---|---|
| TTS | Text-to-Speech |
| HTTP | Hyper Text Transfer Protocol |
| APIs | Application Programming Interfaces |
| JDK | Java Developer Kit |

# CHAPTER 1

# INTRODUCTION

## 1.1 PROBLEM DEFINITION

A number of developing countries continue to provide educational services to students with disabilities in “segregated” schools. Also, all students, regardless of their personal circumstances, have a right of access to and participation in the education system, according to their potential and ability. However, with the rapidly growing population and increasing number of people with blindness along with other disabilities, need for use of technology in the field of education has become imminent. With existing system of competitive examination, students face problems while interacting with the system, misunderstandings arising due to human mediator and also an ability to cope-up with the other students. Our project, through the use of speech technology, attempts to provide solutions for some of these issues by creating an interactive system. Thus, the application will help in creating an environment that provides equal opportunities for all the students in taking up competitive exams. This will improve the current educational system for blinds career.

In the realm of education, the conventional methods of conducting examinations have proven to be inadequate in accommodating individuals with visual impairments, thereby posing significant accessibility challenges. Additionally, educators are burdened with time-consuming administrative tasks associated with manual grading and exam preparation. The emergence of web-based technologies offers a promising avenue to address these challenges and revolutionize the examination process. By harnessing the power of the web, there exists an opportunity to develop an inclusive examination system that caters to the needs of blind individuals while enhancing efficiency and flexibility for all stakeholders.

At the heart of the issue lies the accessibility barrier faced by blind individuals when navigating traditional examination interfaces. These interfaces often rely heavily on visual cues, rendering them inaccessible to those reliant on alternative means of interaction. Furthermore, the administrative burden placed on educators adds another layer of complexity to the examination process, diverting valuable time and resources away from teaching and student support.

# CHAPTER 2

# LITERATURE REVIEW

**Aatisha Cyrill, Shubham Melvin Felix, L. Mary Gladence, [1]** the paper introduces a (TTS) system called "Text Reader for Blind," designed to aid visually impaired individuals. It converts written text into audible speech, enhancing accessibility. The system employs advanced algorithms for accurate pronunciation and natural-sounding speech. It offers customization options and discusses potential applications in education and daily tasks. Overall, the paper highlights the significance of TTS technology in empowering blind individuals and improving their access to information.

**Shagufta Md. Rafique Bagwan, Prof. L.J. Sankpal [2]** The paper introduces "VisualPal," a mobile app developed by Shagufta Md. Rafique Bagwan and Prof. L.J. Sankpal, designed for object recognition to assist visually impaired individuals. Presented at the IEEE International Conference on Computer, Communication, and Control (IC4-2015), VisualPal utilizes smartphone technology to recognize objects in the user's environment in real-time. By leveraging advanced image processing techniques, VisualPal aims to enhance the independence and mobility of visually impaired individuals by providing them with immediate object recognition and assistance. The app's primary goal is to improve the daily lives of visually impaired individuals by facilitating easier navigation and object identification through smartphone-based solutions**.**

**Hanen Jabnoun, Faouzi Benzarti, Hamid Amiri**, **[3]** In their paper presented at the IEEE IPAS'14 conference, Hanen Jabnoun, Faouzi Benzarti, and Hamid Amiri explore object recognition techniques tailored for blind individuals. Their focus is on feature extraction methods, utilizing advanced algorithms to identify objects through image processing. The research aims to develop effective solutions to assist visually impaired individuals in recognizing and interacting with their environment, contributing to improved accessibility and independence for the blind through innovative image processing technologies.

**K. Matusiak, P. Skulimowski and P. Strumiááo,[4]** The paper by K. Matusiak, P. Skulimowski, and P. Strumiński from Lodz University of Technology in Poland presents an

object recognition mobile application tailored for visually impaired users. The study focuses on developing efficient algorithms and integrating them into a smartphone app to aid visually impaired individuals in recognizing objects in their environment. By leveraging mobile technology, the application aims to enhance accessibility and independence for visually impaired users by providing real-time object recognition capabilities. The research contributes to the field of assistive technology by offering practical solutions to address
the challenges faced by visually impaired individuals in daily life.

**Shahed Anzarus Sabab, Md. Hamjajul Ashmafee,[5]** "Blind Reader: An Intelligent Assistant for Blind" by Shahed Anzarus Sabab and Md. Hamjajul Ashmafee, presented at the 19th International Conference on Computer and Information Technology, introduces an intelligent assistant designed for blind individuals. The system, named Blind Reader, utilizes advanced technologies to provide assistance with tasks such as reading text, navigating environments, and accessing information. Through innovative features and functionalities, Blind Reader aims to enhance accessibility and independence for visually impaired individuals, offering practical solutions to address their unique challenges and improve their quality of life.

**Hanen Jabnoun, F aouzi Benzarti, Hamid Amiri [6]"**Object Detection and Identification for Blind People in Video Scene" by Hanen Jabnoun, Faouzi Benzarti, and Hamid Amiri from Université de Tunis EI Manar explores techniques for object detection and identification specifically tailored for blind individuals. Conducted at Ecole Nationale d'Ingénieur de Tunis, the research focuses on developing methods to detect and identify objects within video scenes, leveraging advanced computer vision algorithms. The study aims to provide visually impaired individuals with real-time assistance in recognizing and interacting with their surroundings, contributing to improved accessibility and independence through innovative video processing technologies.

**K.Gopala Krishnan, C.M.Porkodi, K.Kanimozhi,[7]** The paper by K. Gopala Krishnan, C.M. Porkodi, and K. Kanimozhi, presented at the International Conference on Communication and Signal Processing in 2013, discusses "Image Recognition for Visually Impaired People by Sound." The research focuses on developing a system that converts visual information into sound to aid visually impaired individuals in understanding their surroundings. Through innovative image recognition techniques, the system aims to provide real-time auditory feedback, enabling visually impaired individuals to recognize objects and navigate their

environment more effectively. The study contributes to enhancing accessibility and independence for visually impaired individuals through the use of sound-based image recognition technology.

**Akhilesh A. Panchal, Shrugal Varde, M.S. Panse,[8]** "Akhilesh A. Panchal, Shrugal Varde, and M.S. Panse" presented their paper at the IEEE International Conference on Recent Trends in Electronics Information Communication Technology in 2016. The paper discusses a "Character Detection and Recognition System for Visually Impaired People." The system is designed to aid visually impaired individuals in recognizing characters through advanced image processing techniques. By providing real-time character detection and recognition capabilities, the system aims to enhance accessibility and independence for visually impaired individuals in reading and understanding textual information. The research contributes to the development of assistive technologies that address the unique challenges faced by visually impaired individuals in accessing written content.

**Nada N. Saeed, Mohammed A.-M. Salem, Alaa Khamis,[9]** "Nada N. Saeed, Mohammed A.-M. Salem, and Alaa Khamis" from the German University in Cairo and Ain Shams University present a paper on "Android-Based Object Recognition for the Visually Impaired." This research focuses on developing an Android-based system for object recognition to assist visually impaired individuals. By leveraging smartphone technology and advanced image processing algorithms, the system aims to provide real-time object recognition capabilities on mobile devices. The study contributes to enhancing accessibility and independence for visually impaired individuals by offering practical solutions to recognize and interact with objects in their environment.

**Vincent Gaudissart, Silvio Ferreira, Celine Thillou, Bernard Gosselin**,[10] The paper "Mobile Reading Assistant for Blind People" by Vincent Gaudissart, Silvio Ferreira, Celine Thillou, and Bernard Gosselin, presented at the SPECOM'2004 conference, introduces a mobile reading assistant tailored for blind individuals. The system is designed to assist visually impaired individuals in accessing written text by converting it into audible speech. By leveraging mobile technology and speech synthesis techniques, the system aims to enhance accessibility and independence for blind individuals in reading printed materials. The research contributes to the development of assistive technologies that address the unique challenges faced by visually impaired individuals in accessing written content.

**N.G. Bourbakis, D. Kavraki,[11]** "N.G. Bourbakis and D. Kavraki" present "An Intelligent Assistant for Navigation of Visually Impaired People" in 2011. The system is designed to aid visually impaired individuals in navigation by providing real-time assistance and guidance. Leveraging advanced technologies such as GPS and sensors, the intelligent assistant offers audible directions and alerts to help users navigate safely. The research aims to enhance accessibility and independence for visually impaired individuals by offering practical solutions for navigating indoor and outdoor environments.

Noura A. Semary, Sondos M. Fadl, Magda S. Essa, Ahmed F. Gad, **[12]"**Noura A. Semary, Sondos M. Fadl, Magda S. Essa, and Ahmed F. Gad" present a "Currency Recognition System for Visually Impaired" in 2015, focusing on Egyptian banknotes as a case study. The system is designed to assist visually impaired individuals in identifying currency denominations through image processing techniques. By leveraging advanced algorithms, the system provides real-time recognition of Egyptian banknotes, enhancing accessibility and independence for visually impaired individuals in handling financial transactions. The research contributes to addressing the specific needs of visually impaired individuals in recognizing currency, offering practical solutions to facilitate daily financial activities.

# CHAPTER 3

# THEORITICAL BACKGROUND

## 3.1 IMPLEMENTATION ENVIRONMENT

User Interface (UI): For a mobile application, the UI components could be developed using native development frameworks such as Android Studio for Android or Xcode for iOS.

Alternatively, cross-platform development frameworks like React Native or Flutter might be used to build the UI components once and deploy them across multiple platforms.

Mobile Application: The mobile app logic is implemented using programming languages such as Java or Kotlin for Android, the app communicates with the web service using HTTP requests. This can be achieved through built-in libraries or frameworks like Retrofit for Android Web Service: The web service is typically implemented using server-side programming languages like Python (using frameworks like Django or Flask), Node.js (with frameworks like Express.js), Ruby on Rails, or Java (with Spring Boot). The service exposes endpoints (HTTP APIs) that the mobile app can interact with to send requests and receive responses. It handles incoming requests, processes them, interacts with the database as needed, and sends back appropriate responses. This interaction is facilitated through HTTP or other network protocols.

Database: The database could be implemented using relational database management systems (RDBMS) such as MySQL, PostgreSQL, or SQLite. Alternatively, NoSQL databases like MongoDB or Firebase Fire store could be used, depending on the requirements of the application. The database stores the application data in a structured manner and provides mechanisms for querying and retrieving data efficiently. The web service communicates with the database using database-specific libraries or ORMs (Object-Relational Mappers) like Hibernate (for Java) or SQL Alchemy (for Python).

SQL:SQL (Structured Query Language) is used within the web service to interact with the database. SQL queries are constructed dynamically based on the requests received from the mobile application. The SQL queries are executed against the database to retrieve, insert, update, or delete data as required by the application's functionality.

### 3.2 SYSTEM ARCHITECTURE

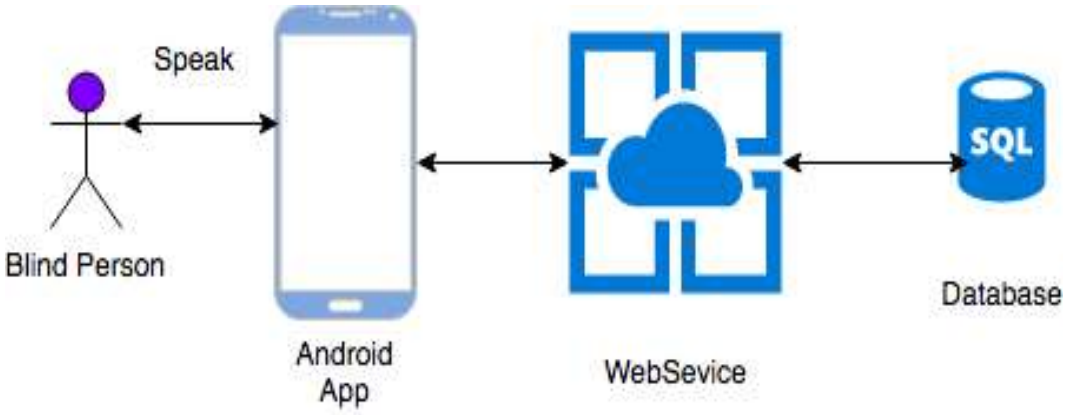



Fig 3.2.1 System Architecture

Client Side (Mobile App): The client side consists of the mobile application running on users' devices (e.g., smartphones or tablets) It provides the user interface (UI) through which users interact with the application. The mobile app is responsible for handling user inputs, displaying information, and facilitating communication with the server side. It communicates with the server side via HTTP requests, sending user queries or commands and receiving responses.

Server Side (Web Service): The server side comprises the web service hosted on a remote server. It handles incoming requests from the client side (mobile app) and processes them to provide the required functionality. The web service acts as an intermediary between the client and the database, executing business logic and managing data retrieval and storage. It exposes APIs that define how clients can interact with the server, typically using HTTP-based protocols like REST (Representational State Transfer)

Database: The database stores and manages the application's data in a structured format. It serves as the backend data store, providing persistent storage for information such as user profiles, preferences, content, etc. The database is accessed by the web service to perform CRUD (Create, Read, Update, Delete) operations in response to client requests. Depending on the application's requirements, the database may use a relational (SQL-based) or non-relational (NoSQL-based) data model. Communication Protocol: Communication between the mobile app and the web service typically occurs over HTTP(S) using RESTful APIs.

## 3.3 PROPOSED METHODOLOGY

This project proposes a system that will create a revolution in a world of education by providing an easier way for visually impaired people to take tests just as normal students do. Rapidly growing population and increasing number of people with blindness along with other disabilities need for use of technology in the field of education has become imminent. With existing system of competitive examination, students face problems while interacting with the system, misunderstandings arising due to human mediator and also an ability to cope-up with the other students. Our project, Android App through the use of speech technology, attempts to provide solutions for some of these issues by creating an interactive system. Thus, the application will help in creating an environment that provides equal opportunities for all the students in taking up competitive exams. This will improve the current educational system for blinds career.

Phone there is a facility which allows a blind people to get interacts with that App as a normal man. If they touch any button or the display option in the Phone has been designed to react as soon as by giving voice output. So, a visor people can use all the facility provided in that Android phone (they are able to access all the options like a normal man).

### Advantages

- They can easily give the answer by voice without any confusion.
- It reduced candidate depressing. The proposed system is user Friendly.
- There is no any need to give the input as manually.
- The blind peoples also can-do quiz exam like a normal human if Test can be taken using an Android Application, Here Questions & choice are through in Voice our project is delivered in real time.
- Blinds can login the exam using their application. The Answer will be getting from the user through voice. The results also delivered through voice. The marks are automatically collected, analyzed, and distributed for purposes like evaluation of teaching and learning process

### 3.4 MODULE DESIGN

#### 3.4.1 System Design

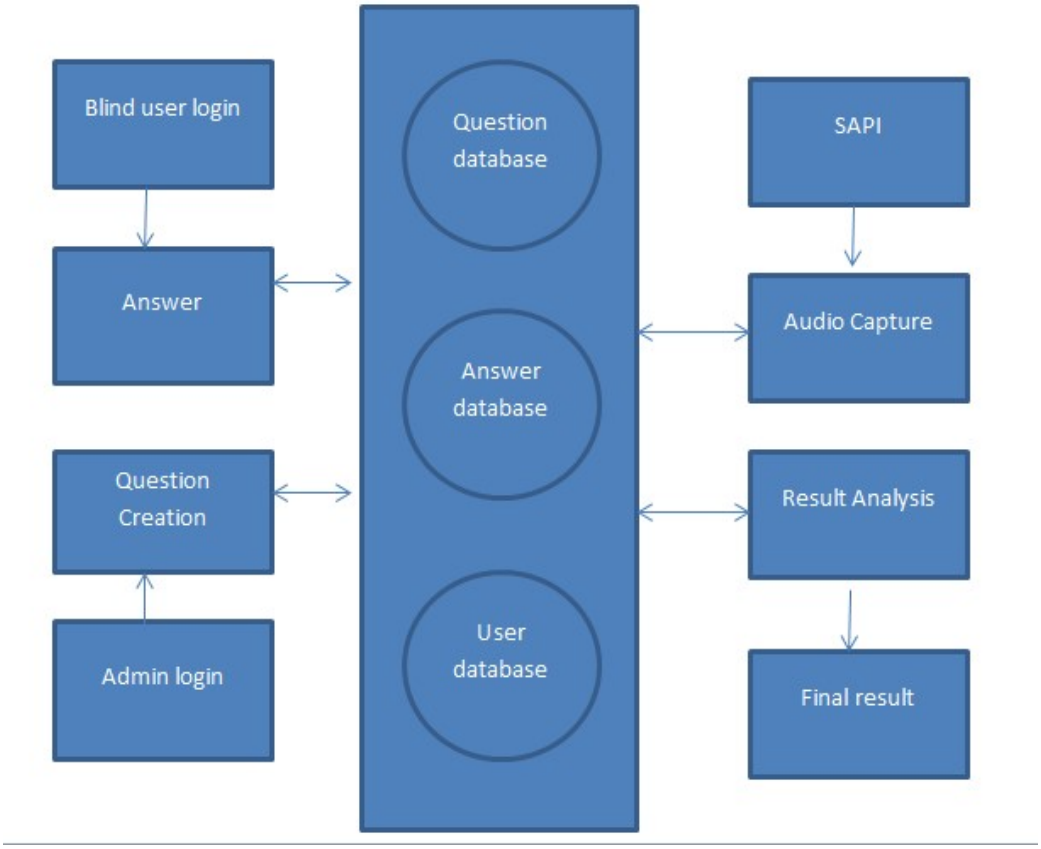



**Fig 3.4.1 System Design**

User Authentication Module: This module handles user registration, login, and authentication processes. Users can create accounts using their email addresses and set up their profiles. Password hashing techniques ensure the security of user credentials. Quiz Management Modulizations users can create, edit, and delete quizzes through this module. Quizzes are categorized based on topics such as algorithms, databases, programming languages, etc. Each quiz contains a set of questions with multiple-choice answers. Question Generation Module: This module generates questions dynamically for each quiz. Questions are fetched from the database based on the selected quiz category and difficulty level. Text-to-Speech (TTS) Module: Enables users to provide answers to quiz questions through voice commands. Utilizes speech recognition algorithms to accurately interpret user responses. Scoring and Evaluation Module: Automatically scores user responses based on the correct answers stored in the database. Generates real-time feedback on user performance and progress.

### 3.4.2 Sequence Diagram

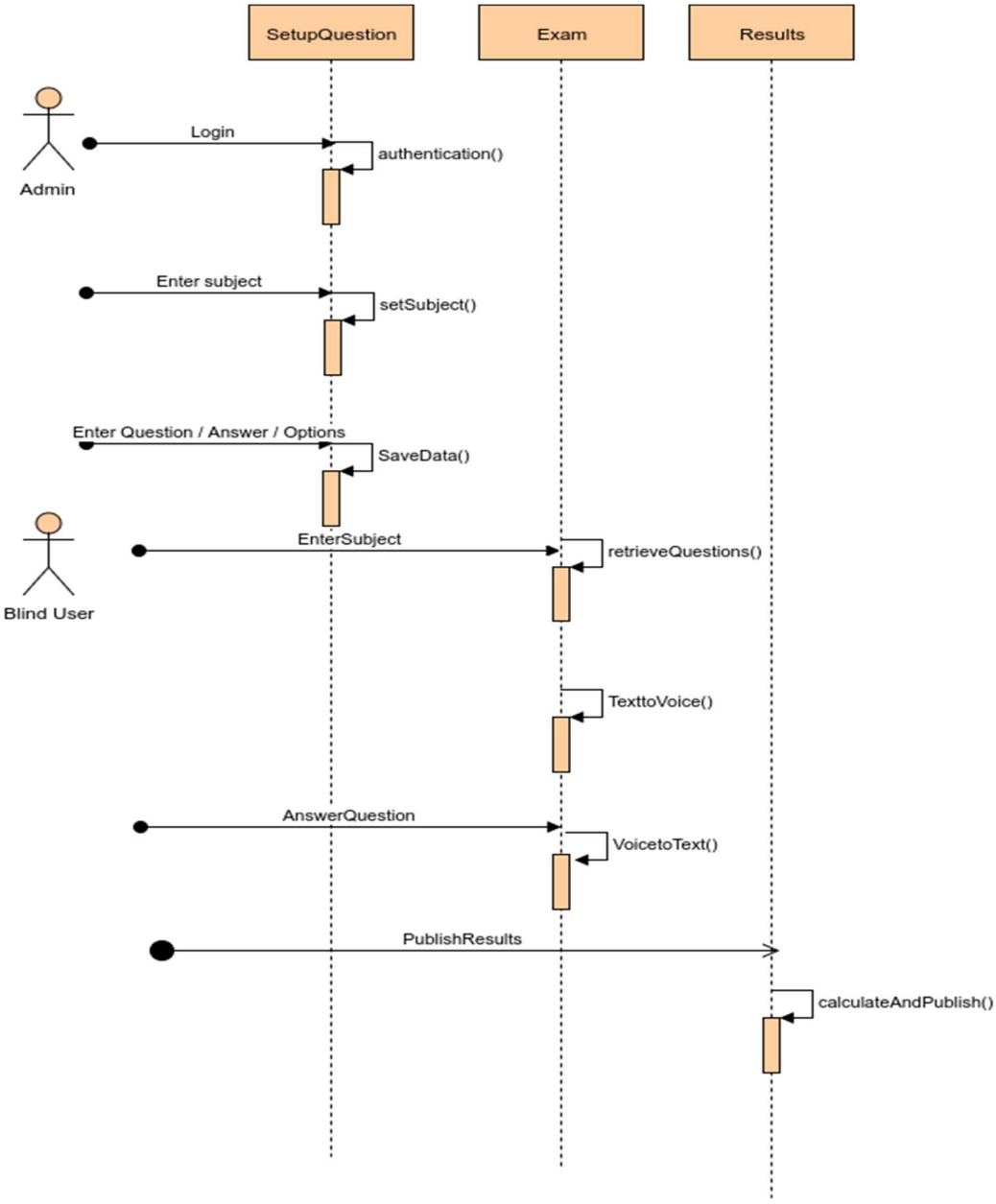



**Fig. 3.4.2 Sequence Diagram**

A Sequence diagram Fig 3.4.2 is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram

### 3.4.3 Use Case Diagram

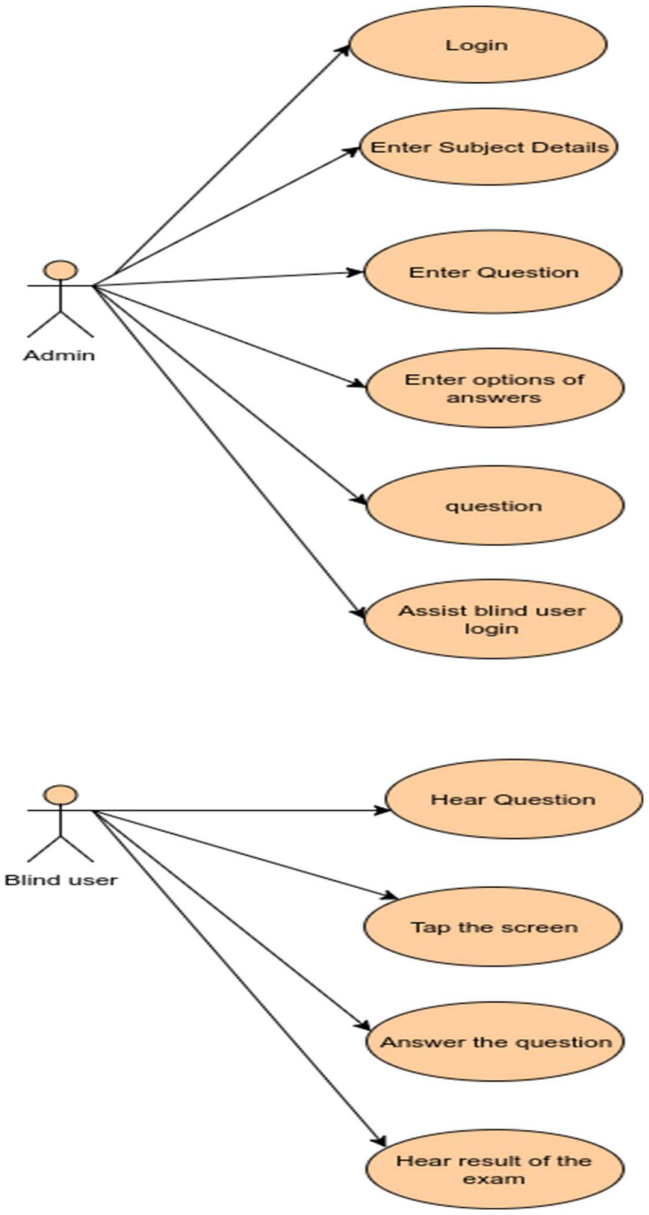



**Fig. 3.4.3 Use Case Diagram**

Unified Modeling Language (UML) Fig.3.4.3.1 is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems

### 3.4.4 Activity Diagram

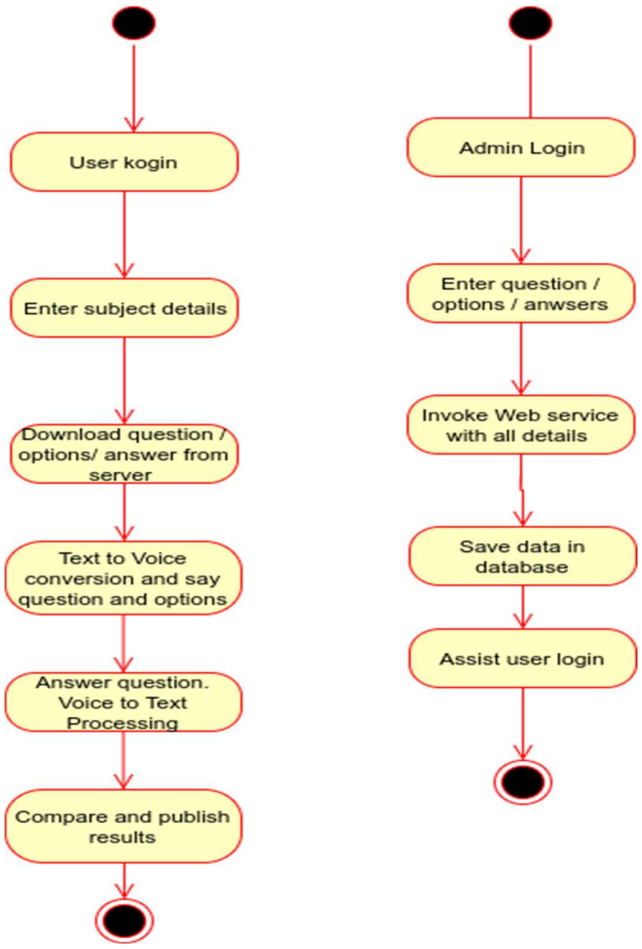



**Fig. 3.4.4 Activity Diagram**

Activity diagram Fig. 3.4.4.1 is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow

### 3.4.6 Class Diagram

Class Diagram Fig.3.4.6.1 is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. Includes a set of graphic notation techniques to create visual models of software intensive systems.

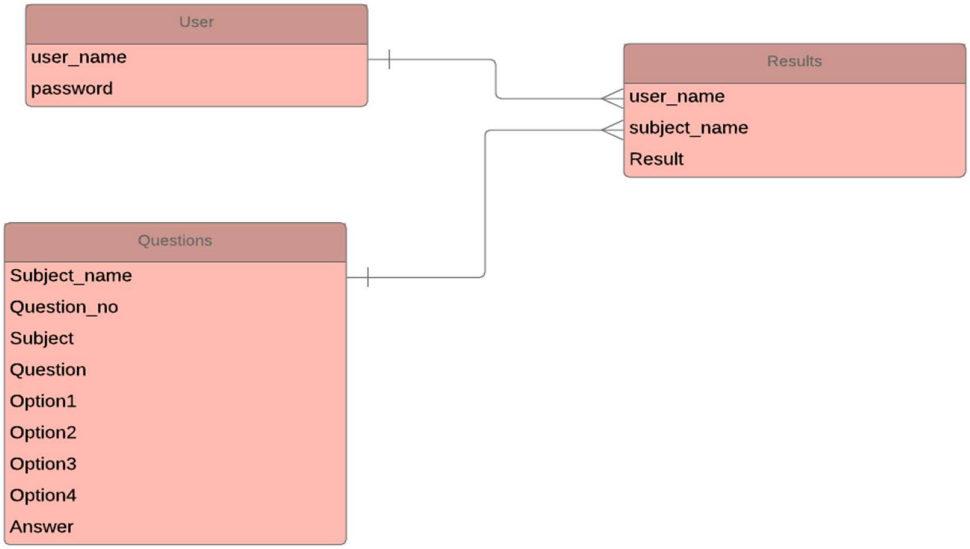



**Fig 3.4.6 Class Diagram**

# CHAPTER 4

# SYSTEM IMPLEMENTATION

For the front end, the Android application will feature a user-friendly interface designed for blind individuals, incorporating voice-activated navigation and integration with accessibility features such as text-to-speech and speech recognition. The interface will allow users to select exams, navigate questions, and provide verbal responses. On the back end, a Python-based cloud server will handle the processing of exam data, including question retrieval, answer validation, and result storage. This server will communicate with the Android application through secure APIs to exchange data and ensure seamless functionality. Key components of the implementation include integrating Google's Text-to-Speech and Speech Recognition APIs for voice interaction, implementing RESTful APIs for communication between the Android app and the Python server, and deploying the Python backend on a cloud platform such as Google Cloud Platform or Amazon Web Services. By combining the strengths of Android Studio for front-end development and Python cloud services for back-end functionality, the system will provide a comprehensive and accessible examination solution for blind individuals, leveraging the power of smartphone technology to promote inclusivity in education.

## 4.1 MODULE EXPLANATION

### 4.1.1 User Interface Module:

This module focuses on designing an intuitive and accessible user interface for the Android application. It includes components for navigation, exam selection, question display, and answer submission. The interface will be optimized for voice interaction and compatible with screen readers.

Voice Interaction Module: This module integrates voice recognition and text-to-speech functionalities to enable seamless interaction between the user and the application. It allows users to navigate through the exam interface, listen to questions, and provide verbal responses.

Exam Management Module: This module handles the management of exams, including storing exam data, retrieving questions, and organizing exam sessions. It ensures that exams are

accessible and properly formatted for blind individuals, taking into account factors such as font size, contrast, and layout.

Communication Module: This module facilitates communication between the Android application and the Python-based cloud server. It includes APIs for exchanging data, such as exam responses and results, securely over the internet.

Backend Processing Module: This module operates on the Python cloud server and handles the processing of exam data. It includes components for validating answers, storing exam results, and generating feedback for users. It ensures the integrity and security of the examination process.

Authentication and Security Module: This module manages user authentication and implements security measures to protect sensitive exam data. It includes features such as user login, session management, and encryption of data transmitted between the client and server.

Configuration and Customization Module: This module allows users to customize their exam experience according to their preferences and needs. It includes options for adjusting settings such as speech rate, voice preferences, and interface layout.

By modularizing the implementation in this way, each component can be developed and tested independently, allowing for easier maintenance and scalability of the system. Additionally, clear separation of concerns between the front end and back-end modules ensures a more robust and flexible architecture.

### 4.1.2 My Notes

**User-Centric Design**

Prioritize user experience and accessibility throughout the development process. Ensure that the application is intuitive and easy to navigate for blind individuals, leveraging voice interaction and other accessibility features.

**Voice Interaction**

Implement robust voice recognition and text-to-speech functionalities to enable seamless interaction with the application. Test the voice interaction features thoroughly to ensure accuracy and reliability.

**Security Considerations**

Implement strong security measures to protect sensitive exam data, including user authentication, encryption of data in transit, and secure storage of user information.

**Compatibility and Integration**

Ensure compatibility with various Android devices and versions, and test the application across different screen sizes and resolutions. Integrate seamlessly with external APIs and services for voice processing and cloud-based backend functionalities.

**Scalability and Performance**

Design the application architecture with scalability and performance in mind to accommodate potential growth in user base and exam volume. Optimize the application for efficient use of system resources and minimize latency in communication with the backend server.

**Testing and Quality Assurance**

Conduct thorough testing of the application to identify and address any bugs or issues. Test for accessibility compliance and usability with blind individuals, gathering feedback for iterative improvements.

**Documentation and Support**

Provide comprehensive documentation for users and developers, including user guides and technical documentation for API usage. Offer ongoing support and maintenance to address any issues or concerns that arise post-launch.

**Compliance and Regulations**

Ensure compliance with relevant regulations and standards, such as accessibility guidelines (e.g., WCAG) and data protection regulations (e.g., GDPR). Adhere to best practices for data privacy and security throughout the development lifecycle.

**Feedback and Iteration**

Solicit feedback from blind individuals and stakeholders throughout the development process to ensure that the application meets their needs and expectations. Iterate on the design and functionality based on feedback received.

**Community Engagement**

Engage with the blind community and relevant organizations to raise awareness of the application and gather insights for continuous improvement. Collaborate with experts in accessibility and assistive technology to enhance the application's effectiveness for blind individuals.

### 4.1.3 Voice Operated Tool- Examination Portal for Blind Persons

Speech recognition applications are becoming more and more useful now a days. Various interactive speech aware applications are available in the market. But they are usually meant for and executed on the traditional general-purpose computers. With growth in the needs for embedded computing and the demand for emerging embedded platforms, it is required that the Speech Recognition System (SRS) are available on them too. In this paper, we are presenting a desktop application named as Examination Portal for Blind Persons so as to eliminate the use of an assistant while giving a multiple-choice question type examination. Further this tool can be implemented as a mobile application and a web-based application.

**Speech Recognition** (SR) is the translation of spoken words into text. It is also known as "Automatic Speech Recognition" (ASR), "Computer Speech Recognition" (CSR) or just "Speech to Text" (STT). Speech Recognition is not providing ease in accessing such systems to only blind people but also who have partial vision and differently able people. In last few years due to continuous incremental improvements, speech recognition has become a strong medium to translate spoken words into texts. Various software's like Google Voice and Siri have allowed mobile users to just use their voice to operate their cell phones.

**Examination System** for Blinds with Real Time Voice Interface: Examinations are the basic building block for academic progress. Person gets promoted to higher level or completes a course once they complete some tests. 80% of competitive examinations in India are of objective type. They generally contain 4 or 5 optional questions. Reading questions during examinations are really tough for the blind or semi blind people due to their limited visibility. Being software engineers, we hold responsibility to help those people.

I propose to develop a complete system which solves all these problems in real time by reading questions written in during examination. Even it reads the objective answers. Audio can be listened for headphone. We used SAPI tool inbuilt with the VB to convert text to audio. The system will be verified for its efficiency in real time. Even for future enhancement we have ideas to add voice commands activated like ―read again‖, ―next page‖, _ended. We hope our

system provide the blind people efficiency to tackle their examination related problems in real time.

**Voice Based Online Examination for Physically Challenged**: A number of developing countries continue to provide educational services to students with disabilities in "segregated" schools. Also, all students, regardless of their personal circumstances, have a right of access to and participation in the education system, according to their potential and ability. However, with the rapidly growing population and increasing number of people with blindness along with other disabilities, need for use of technology in the field of education has become imminent.

With existing system of competitive examination, students face problems while interacting with the system, misunderstandings arising due to human mediator and also an ability to cope-up with the other students. Our project, through the use of speech technology, attempts to provide solutions for some of these issues by creating an interactive system. Thus, the application will help in creating an environment that provides equal opportunities for all the students in taking up competitive exams. This will improve the current educational system for blinds career.

The growth of the Internet, and in particular the World Wide Web, is already influencing the way science is taught and will undoubtedly do so to greater extent in the future. In areas of education, it offers a medium that has the potential to be more responsive to students. To encourage greater participation in their own learning, and to give greater access to different sources of information than traditional method offers. In the future blind peoples also can-do online exam like a normal human if our project is delivered in real time.

University Examination System for Students with Visual Impairments: This paper presents the development of a web based, platform independent system for university examination purposes that can be easily accessed and used by students with visual impairments, with minimum effort required to learn its use. The developed examination system allows students with visual impairments to take suitably adapted online written examinations according to their individual and personalized special characteristics and preferences for reading digital text.

# CHAPTER 5

# RESULTS AND DISCUSSION

## 5.1 TESTING

Once the design aspect of the system is finalizing the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system

| TEST CASE ID | TESTCASE/ ACTIONTOBE PERFORMED | EXPECTED RESULT | ACTUAL RESULT | PASS/ FAIL |
|---|---|---|---|---|
| 1. | TAPP “TO SPEAK” button | Display to voice recognition | Display to voice recognition | Pass |
| 2. | Selecting take test button | Display test page | Display test page | Pass |
| 3. | Select “create test exam” button | Enter into To exam page | Enter into To exam page | Pass |
| 4. | Select “login” Button | Enter into login page | Enter into login page | Pass |
| 5. | Selecting “Advanced Test | Display the details and, etc., | Display the details and, etc., | Pass |

| | | | | |
|---|---|---|---|---|
| 6 | Selecting "CANCEL" | Display cancel new request, in progress request. | Displays cancel new request, in progress request. | Pass |
| 7. | Clicking the "ADD Exam" button | Add Test to backend | Add Test to backend | Pass |
| 8. | Read text of the test | Read text | Read text | Pass |
| 9. | Getting "AUDIO RESPONSE" button | Test question audio details | Test question audio details | Pass |
| 10. | Select "SIGNOUT" button | Logs out the user | Logs out the user | Pass |
| 11. | Select "Take Test" Button | Show the question page and book to select | Show the question page and book to select | pass |
| 12. | Select "CREATE NEW TEST" button | Show the create note button and works properly | Show the create note button and works properly | pass |
| 13 | Display result and read aloud | Read the text | Read the text | pass |

## CODING STANDARDS

Coding standards are guidelines that define the style, structure, and conventions to be followed while writing code. These standards ensure consistency, readability, and maintainability across software projects. Key aspects of coding standards include: Naming Conventions: Consistent naming of variables, functions, classes, and other elements improves code clarity and comprehension. Descriptive and meaningful names should be used to convey the purpose and functionality of each entity. Formatting Rules: Consistent indentation, spacing, and line wrapping enhance code readability and organization. Adopting a standardized format improves code consistency and makes it easier for developers to understand and navigate the codebase. Documentation Practices: Clear and concise documentation, including comments, function/method headers, and README files, helps developers understand the code's functionality and usage. Documenting code effectively facilitates collaboration and maintenance.

Error Handling: Implementing consistent error handling mechanisms improves code robustness and reliability. Error messages should be informative and actionable, aiding in troubleshooting and debugging. Code Modularity and Reusability: Encapsulation and modularization of code promote reusability and maintainability. Breaking down code into smaller, reusable components enhances scalability and facilitates code maintenance. Performance Optimization: Adhering to performance optimization techniques and best practices improves code efficiency and resource utilization. Considerations such as algorithm complexity, memory management, and I/O operations impact application performance.

Security Measures: Implementing secure coding practices helps mitigate vulnerabilities and protect against potential security threats. Avoiding common security pitfalls such as SQL injection, cross-site scripting, and buffer overflows enhances application security. Version Control and Collaboration: Following version control best practices, such as meaningful commit messages and branching strategies, facilitates collaboration and enables effective code management.

Continuous Improvement: Regularly reviewing and updating coding standards based on feedback and evolving industry practices fosters continuous improvement and ensures the long-term viability of the codebase.

## 5.2 RESULTS

**Technology Integration**

- Android Frontend: Leveraging Android Studio, create an intuitive interface with voice interaction for blind users to access exams seamlessly. Python Backend on Cloud: Utilizing Python cloud services, implement robust backend processing for exam data, including validation and result storage.
- Voice-to-Text and Text-to-Voice Integration: Integrate advanced voice recognition and text-to-speech features to enable blind individuals to listen to questions and provide verbal responses effortlessly.

**Inclusive Curriculum Development**

- Collaborate with educators, content creators, and accessibility experts to adapt existing educational materials and develop new, inclusive curriculum content. Ensure that the curriculum covers a wide range of subjects and is tailored to various educational levels.

**Teacher Training and Support**

- Provide training programs for educators on best practices for teaching visually impaired students. - Establish a support system for teachers, including resources and professional development opportunities.

**Community Engagement**

- Foster a sense of community among visually impaired students through mentorship programs

**Educational Achievement**

- Track the academic performance of visually impaired students participating in the program compared to previous years.

**User Satisfaction**

- Collect feedback from students, teachers, and parents to assess the satisfaction levels with the project's initiatives.

**Inclusion Metrics**

- Monitor the increased enrollment and retention rates of visually impaired students in mainstream educational settings.

## 5.3 DISCUSSION

**Understanding User Needs:** Prioritizing the understanding of the needs and challenges faced by blind individuals is paramount. Conducting thorough user research and engaging directly with blind users ensures that the application addresses their specific requirements effectively.

**Accessibility as a Priority:** Accessibility should be at the forefront of design considerations, ensuring that the application is usable by individuals with various levels of visual impairment. Implementing features such as voice interaction, high contrast modes, and screen reader compatibility guarantees inclusivity for all users.

**Customization and Personalization:** Recognizing the diverse preferences and abilities of blind users, providing customization options within the application is essential. Allowing users to personalize settings such as speech rate, voice preferences, and interface layout enhances usability and accommodates individual needs.

**Seamless Navigation and User Experience:** Designing an intuitive and streamlined navigation experience is crucial for ensuring ease of use. Users should be able to navigate through the application effortlessly, accessing exams, questions, and features with minimal friction or confusion.

**Integration of Advanced Technologies:** Leveraging advanced technologies such as voice recognition, text-to-speech, and cloud-based processing enhances the functionality and effectiveness of the application. Integrating these technologies seamlessly into the user experience improves accessibility and usability for blind individuals.

**Continuous Improvement and User Feedback:** Emphasizing continuous improvement through user feedback loops fosters ongoing enhancements to the application. Soliciting feedback from blind users and stakeholders allows for iterative improvements and ensures that the application evolves to meet changing needs and preferences.

**Community and Support Networks** munity and support networks are vital for the success of the Android-based examination solution for blind individuals. These networks provide

forums, advocacy groups, and developer communities for collaboration, resource sharing, and assistance. Through these channels, users receive support, guidance, and a sense of belonging, enhancing accessibility and fostering a supportive ecosystem.

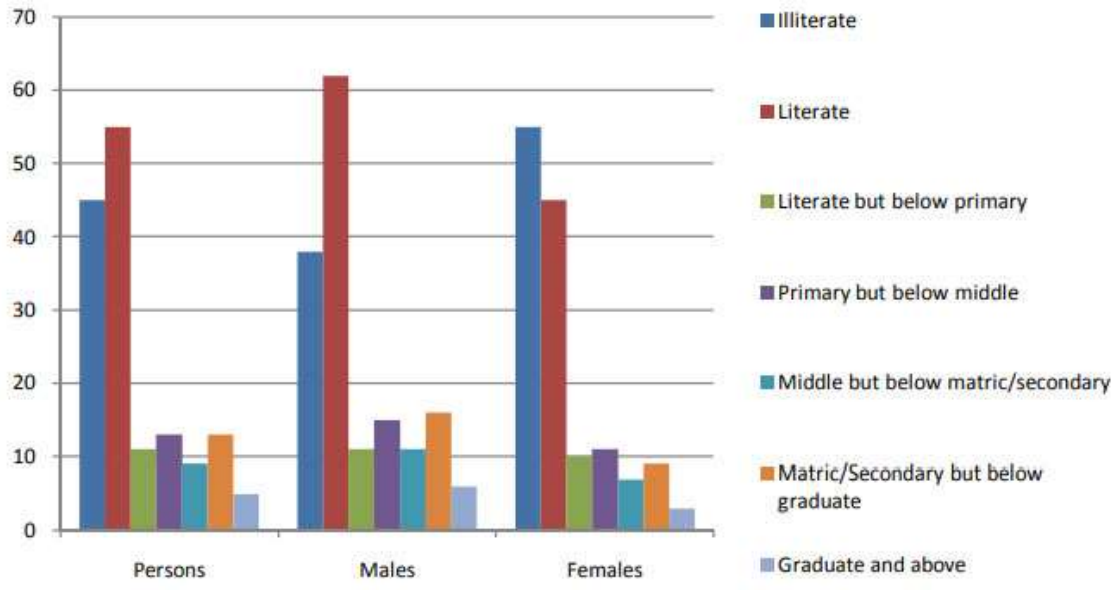



Fig 5.3.1 Number of blind people who is illiterate

In fig5.3.1 show the increasing number of illiterate blind individuals due to the lack of online testing presents a significant challenge in ensuring equal access to education and opportunities for this demographic. Without accessible online tests tailored to their needs, illiterate blind individuals face barriers in assessing their knowledge, skills, and abilities, limiting their participation in educational programs, employment opportunities, and social inclusion Exact statistics on the number of blind individuals who are illiterate in different educational levels may vary by region and are not readily available. A significant challenge globally. In primary education: Illiteracy among blind individuals in primary school may be more prevalent due to various factors such as limited access to inclusive education, insufficient resources for specialized instruction, and societal stigmas surrounding disability. In high school: The number of illiterate blind individuals in high school could be lower compared to primary education, as some may have received basic literacy training or accommodations in earlier grades. However, challenges persist, including inaccessible learning materials and curriculum barriers. In graduate education: While fewer blind individuals may reach graduate education levels, those who do face unique challenges related to academic ragout, access to specialized

# CHAPTER 6

# CONCLUSION & FUTURE WORK

## 6.1 CONCLUSION

In conclusion, the development of an Android-based examination solution for blind individuals represents a significant stride towards fostering inclusivity and accessibility in education. By leveraging smartphone technology and integrating advanced features like voice interaction and cloud-based processing, this solution aims to empower blind individuals to participate in exams independently and confidently. Through prioritizing user needs, accessibility, and continuous improvement, this project endeavors to provide a seamless and personalized experience for blind users. By understanding the challenges faced by blind individuals and actively involving them in the development process, the solution can better cater to their unique requirements and preferences. Moreover, the establishment of community and support networks further enhances the success and sustainability of the solution. By fostering collaboration, resource sharing, and advocacy, these networks create a supportive ecosystem where users can receive assistance, guidance, and a sense of belonging. In essence, the Android-based examination solution for blind individuals not only addresses a critical need in the education sector but also embodies the spirit of innovation, inclusivity, and empowerment. By breaking down barriers and providing equal opportunities for blind individuals to pursue their educational goals, this solution has the potential to make a profound impact on the lives of countless individuals worldwide.

## 6.2 FUTURE ENHANCEMENTS

**Enhanced Accessibility Features**: Continuously improving accessibility features such as voice recognition accuracy, text-to-speech naturalness, and compatibility with assistive technologies to further enhance usability for blind individuals.

**Integration of AI and Machine Learning**: Leveraging AI and machine learning algorithms to provide personalized learning experiences, adaptive question generation, and real-time feedback based on user performance and preferences.

**Expanded Content and Subject Coverage**: Adding support for a wider range of subjects, exam formats, and question types to accommodate diverse educational needs and disciplines.

**Multi-platform Compatibility**: Extending compatibility to other platforms beyond Android, such as iOS, web browsers, and desktop applications, to reach a broader audience of users.

**Gamification and Engagement Features**: Introducing gamification elements, rewards, and interactive features to increase user engagement, motivation, and retention during exam preparation and participation.

**Enhanced Security Measures**: Implementing advanced security measures to safeguard exam integrity, prevent cheating, and ensure compliance with regulatory standards and best practices.

**Integration with Learning Management** Systems: Integrating seamlessly with learning management systems (LMS) and educational platforms to streamline exam administration, data management, and reporting processes for educators and administrators.

**Collaboration and Social Learning Tools**: Incorporating collaborative features such as group study sessions, peer-to-peer learning networks, and social forums to facilitate knowledge sharing and community building among blind individuals.

**Localization and Multilingual Support**: Providing support for multiple languages and localization features to cater to diverse linguistic and cultural backgrounds among users worldwide.

**User Feedback and Iterative Development**: Continuously gathering user feedback, conducting usability testing, and iterating on the application based on insights and suggestions from blind individuals, educators, and stakeholders to ensure ongoing improvement and relevance to user needs

.

## REFERENCES


[1] Aatisha Cyrill, Shubham Melvin Felix, L. Mary Gladence, "Text Reader for Blind: Text-To-Speech", International Journal of Pure and Applied Mathematics Volume 117 No. 21, 119-125, 2017.

[2] Shagufta Md. Rafique Bagwan, Prof. L.J. Sankpal," VisualPal: A Mobile App for Oject Recognition for the Visually Impaired", IEEE International Conference on Computer, Communication and Control (IC4-2015).

[3] Hanen Jabnoun, Faouzi Benzarti, Hamid Amiri, "Object recognition for blind people based on features extraction", IEEE IPAS'14: International Image Processing Applications and Systems Conference 2014.

[4] K. Matusiak, P. Skulimowski and P. Strumiááo," Object recognition in a mobile phone application for visually impaired users", Lodz University of Technology, Lodz, Poland.

[5] Shahed Anzarus Sabab, Md. Hamjajul Ashmafee, "Blind Reader: An IntelligentAssistant for Blind", 19th International Conference on Computer and Information Technology, December 18-20, 2016, North South University, Dhaka, Bangladesh.

[6] Hanen Jabnoun, F aouzi Benzarti, Hamid Amiri, "Object Detection and Identification for Blind People in Video Scene", Universite de Tunis EI Manar, Ecole Nationale d'Ingenieur de Tunis 1002, Tunis Le Belvedere, Tunisie.

[7] K. Gopala Krishnan, C.M. Porkodi, K. Kanimozhi, "Image Recognition for Visuall Impaired People by Sound", International conference on Communication and Signal Processing, April 3-5, 2013, India

[8] Akhilesh A. Panchal, Shrugal Varde, M.S. Panse," Character Detection and Recognition System for Visually Impaired People", IEEE International Conference on Recent Trends in Electronics Information Communication Technology, May 20-21, 2016, India.

[9] Nada N. Saeed, Mohammed A.-M. Salem, Alaa Khamis," AndroidBased Obect Recognition for the Visually Impaired", German University in Cairo, Ain Shams University

[10] Vincent Gaudissart, Silvio Ferreira, Celine Thillou, Bernard Gosselin, “Mobile Reading Assistant for Blind People”, SPECOM’2004: 9th Conference Speech and Computer St. Petersburg, Russia September 20- 22, 2004.

[11] N.G. Bourdais, D. Kavraki,” An Intelligent Assistant for Navigation of Visually Impaired People”, 2011.

[12] Noura A. Semary, Sondos M. Fadl, Magda S. Essa, Ahmed F. Gad,” Currency Recognition System for Visually Impaired: Egyptian Banknote as a Study Case”, Menoufia, Egypt, 2015

# APPENDICES

## A.1 SOURCE CODE

```
package com.mirror.blindquizv2;

import android.app.Activity;
import android.app.Dialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class BQAuthenticationActivity extends Activity implements OnClickListener {
    /**
     * Called when the activity is first created.
     */

    private EditText medit_username;
    private EditText medit_password;
    private Button mbutton_submit;

    String user_name;
    String password;



    Dialog dialog;
    String ip;
```

```
@Override
public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.authentication);



   medit_username = (EditText) findViewById(R.id.edit_user);
   medit_password = (EditText) findViewById(R.id.edit_password);
   mbutton_submit = (Button) findViewById(R.id.button_submit);

   mbutton_submit.setOnClickListener(this);



}

@Override
public void onClick(View arg0) {
   // TODO Auto-generated method stub

   System.out.println("UserName :" + medit_username.getText().toString());
   System.out.println("Password :" + medit_password.getText().toString());
   user_name = medit_username.getText().toString();
   password = medit_password.getText().toString();

   //   Bundle bundle = new Bundle();
   //   bundle.putString("user", user);
   //  bundle.putString("pass", pass);

   if (user_name.equals("admin") && password.equals("admin")) {

      Intent l = new Intent(this, BQUserOptionsActivity.class);
      startActivity(l);
```

```
        } else {


            Toast.makeText(this, "Incorrect UserName or Password",
Toast.LENGTH_SHORT).show();

            //Intent l = new Intent(this,ScanItemActivity.class);
            //startActivity(l);

                        }


    }


}

package com.mirror.blindquizv2;

import android.Manifest;
import android.app.Activity;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
```

```
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ScrollView;
import android.widget.Toast;

import androidx.core.app.ActivityCompat;

import com.android.volley.DefaultRetryPolicy;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.RetryPolicy;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.Locale;

public class BQExamActivity extends Activity implements OnClickListener {
    /**
     * Called when the activity is first created.
     */

    private EditText medit_subject;
    private EditText medit_qno;
    private EditText medit_question;
    private EditText medit_optiona;
    private EditText medit_optionb;
```

```
private EditText medit_optionc;
private EditText medit_optiond;
private EditText medit_answer;

TextToSpeech t1;

ProgressDialog progressDialog;

boolean rec = false ;
boolean flag_first = true;

private final int REQ_CODE_SPEECH_INPUT = 100;


private LinearLayout ml1;
private LinearLayout ml2;
private ScrollView ms1;

private Button mbutton_submit;

JSONArray res;

String subject;
String qno;
String question;
String optiona;
String optionb;
String optionc;
String optiond;
String answer;

String s;
```

```
int q = 0;
int no_correct_answers = 0;



Dialog dialog;
String ip;

@Override
public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.exam);

   ml1 = (LinearLayout) findViewById(R.id.linear1);
   ml2 = (LinearLayout) findViewById(R.id.linear2);

   ms1 = (ScrollView) findViewById(R.id.scroll1);

   ml1.setOnClickListener(this);
   ml2.setOnClickListener(this);
   ms1.setOnClickListener(this);

   Bundle extras = getIntent().getExtras();
   s = (String) extras.getString("subject");



   medit_subject = (EditText) findViewById(R.id.edit_subject);
   medit_qno = (EditText) findViewById(R.id.edit_qno);
   medit_question = (EditText) findViewById(R.id.edit_question);
   medit_optiona = (EditText) findViewById(R.id.edit_optiona);
   medit_optionb = (EditText) findViewById(R.id.edit_optionb);
   medit_optionc = (EditText) findViewById(R.id.edit_optionc);
   medit_optiond = (EditText) findViewById(R.id.edit_optiond);
```

```
medit_subject.setEnabled(false);
medit_question.setEnabled(false);
medit_qno.setEnabled(false);
medit_question.setEnabled(false);
medit_optiona.setEnabled(false);
medit_optionb.setEnabled(false);
medit_optionc.setEnabled(false);
medit_optiond.setEnabled(false);

t1=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
   @Override
   public void onInit(int status) {
      if (status == TextToSpeech.SUCCESS) {

         int result = t1.setLanguage(Locale.US);

         if (result == TextToSpeech.LANG_MISSING_DATA
               || result == TextToSpeech.LANG_NOT_SUPPORTED) {
            Log.e("TTS", "This Language is not supported");
         } else {
            //btnSpeak.setEnabled(true);
            //speakOut();
         }

      } else {
         Log.e("TTS", "Initilization Failed!");
      }
   }

});

//retrieveQuestion();
getQuestions();
```

```
}

public void speakQuestion(){

   try{
      System.out.println("In Speak Question");
      q++;
      boolean next_question = false;
      for (int i=0; i<res.length(); i++) {
         JSONObject obj = res.getJSONObject(i);

          qno = obj.getString("qno");
          subject = obj.getString("subject");
          question = obj.getString("question");
          optiona = obj.getString("optiona");
          optionb = obj.getString("optionb");
          optionc = obj.getString("optionc");
          optiond = obj.getString("optiond");

         answer = obj.getString("answer");


         System.out.println("Subject : " + subject) ;
         System.out.println("Question : " + qno) ;
         System.out.println("Q : " + q) ;

         if (subject.equals(s) && Integer.parseInt(qno) == q){
```

```
                next_question = true;
                medit_subject.setText(subject);
                medit_qno.setText(qno);
                medit_question.setText(question);
                medit_optiona.setText(optiona);
                medit_optionb.setText(optionb);
                medit_optionc.setText(optionc);
                medit_optiond.setText(optiond);
                String text;
               if(rec || flag_first){
                text    = "Question. " + question  + " \n Option A \n " + optiona  + " \n Option B
\n " + optionb  + " \n Option C \n " + optionc +  "\n Option D \n " + optiond ;
                   flag_first = false;
               }
                else {
                    text = "Answer not recognized. Repearting question \n + Question. " +
question  + " \n Option A \n " + optiona  + " \n Option B \n " + optionb  + " \n Option C \n " +
optionc +  "\n Option D \n " + optiond ;
               }


                t1.speak(text, TextToSpeech.QUEUE_FLUSH, null);


            }

        }

        if(!next_question){

            System.out.println("No more questions");
            System.out.println("No.of correct answers :" + no_correct_answers);
            q--;
            System.out.println("Total no of questions :" + q);
```

```
            Bundle extras = new Bundle();
            extras.putString("total",q+"");
            extras.putString("correct",no_correct_answers+"");

            Intent i = new Intent(this, BQMainActivity.class);
            i.putExtras(extras);
            startActivity(i);
        }

        }catch(Exception e) {
        e.printStackTrace();
    }



}

public void getQuestions(){

    RequestQueue queue = Volley.newRequestQueue(this);


    JSONObject req = new JSONObject();
    try {
        req.put("subject", s);

        //  req.put("last_name", lastname);
    } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    String url ="https://d83arun.pythonanywhere.com/getQuestions";
```

```
        System.out.println("URL :" + url);

        JsonObjectRequest jsonobj = new JsonObjectRequest(Request.Method.POST, url,req,
                new Response.Listener<JSONObject>() {
                    @Override
                    public void onResponse(JSONObject response) {

                        Log.d("Chatbot","Success response");
                        System.out.println("Response: " + response);
                        progressDialog.dismiss();

                        try {
                            if((response.getString("response")).equals("success")){
                                // sent_otp = response.getString("otp");
                                // System.out.println("OTP: " + sent_otp);
                                Toast.makeText(getApplicationContext(),"Question retrieved
successfully",Toast.LENGTH_SHORT).show();

                                String questions_res = response.getString("questions");
                                res = new JSONArray(questions_res);
                                System.out.println(res);
                                speakQuestion();
                                //Toast.makeText(getApplicationContext(),"Login with registered mobile
no",Toast.LENGTH_SHORT).show();
                                //finish();
                            /*    medit_subject.setText("");
                                medit_qno.setText("");
                                medit_question.setText("");
                                medit_optiona.setText("");
                                medit_optionb.setText("");
                                medit_optionc.setText("");
                                medit_optiond.setText("");
                                medit_answer.setText(""); */
```

```
                        // mregister.setVisibility(View.VISIBLE);
                        // mbuttonotp.setVisibility(View.INVISIBLE);
                        //((Activity)getApplicationContext()).finish();
                        //Intent i = new
Intent(getApplicationContext(),RegistrationActivity.class);
                        //startActivity(i);

                    }else{
                        Toast.makeText(getApplicationContext(),"Error while Registering
User",Toast.LENGTH_SHORT).show();

                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }



            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                error.printStackTrace();
                progressDialog.dismiss();
                Toast.makeText(getApplicationContext(),"Error while
connecting",Toast.LENGTH_SHORT).show();

            }
        }
    ){
        //here I want to post data to sever
```

```
        };

        System.out.println("In 1");
        // Add the request to the RequestQueue.
        int socketTimeout = 10000;//2 minutes - change to what you want
        //RetryPolicy policy = new DefaultRetryPolicy(socketTimeout,
DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
DefaultRetryPolicy.DEFAULT_BACKOFF_MULT);
        RetryPolicy policy = new DefaultRetryPolicy(socketTimeout, 0,
DefaultRetryPolicy.DEFAULT_BACKOFF_MULT);
        jsonobj.setRetryPolicy(policy);
        queue.add(jsonobj);

        System.out.println("In 2");
        //initialize the progress dialog and show it
        progressDialog = new ProgressDialog(this);
        progressDialog.setMessage("Adding question");
        progressDialog.show();


    }

    public void retrieveQuestion(){


        ActivityCompat.requestPermissions(this,new
String[]{Manifest.permission.INTERNET}, 1);



        RequestQueue queue = Volley.newRequestQueue(this);


        String url ="http://mirrortech.in/quiz/register.php?pass=arun&type=get&subject=test3" ;
```

```
        System.out.println("URL : " + url);
        url = url.replaceAll(" ", "%20");

// Request a string response from the provided URL.
        StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        // Display the first 500 characters of the response string.
                        // mTextView.setText("Response is: "+ response.substring(0,500));



                        Toast.makeText(BQExamActivity.this,"User Data received successfully",
Toast.LENGTH_SHORT).show();
                        System.out.println("Response :" + response);
                        try{
                            res =  new JSONArray(response);
                         /*   for (int i=0; i<res.length(); i++) {
                                JSONObject obj = res.getJSONObject(i);

                                name = obj.getString("name");
                                email = obj.getString("email");
                                dob = obj.getString("dob");
                                gender = obj.getString("gender");



                                if (name.equals(user_name)){

                                    medit_name.setText(name);
                                    medit_email.setText(email);
                                    medit_dob.setText(dob);
                                    medit_gender.setText(gender);
```

```
                    }

                } */

                    speakQuestion();
                }catch (Exception e){
                    e.printStackTrace();
                }

            }    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            System.out.println("Error in getting response");
            error.printStackTrace();
        }
    });

    queue.add(stringRequest);


}


@Override
public void onClick(View arg0) {
    // TODO Auto-generated method stub

    System.out.println("OnClick");

    promptSpeechInput();
}
```

```
private void promptSpeechInput() {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT,
            getString(R.string.speech_prompt));
    try {
        startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
    } catch (ActivityNotFoundException a) {
        Toast.makeText(getApplicationContext(),
                getString(R.string.speech_not_supported),
                Toast.LENGTH_SHORT).show();
    }
}


 /*   subject = medit_subject.getText().toString();
    qno = medit_qno.getText().toString();
    question = medit_question.getText().toString();
    optiona = medit_optiona.getText().toString();
    optionb = medit_optionb.getText().toString();
    optionc = medit_optionc.getText().toString();
    optiond = medit_optiond.getText().toString();
    answer = medit_answer.getText().toString(); */
    //   Bundle bundle = new Bundle();
    //   bundle.putString("user", user);
    //  bundle.putString("pass", pass);
```

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
   super.onActivityResult(requestCode, resultCode, data);

   switch (requestCode) {
      case REQ_CODE_SPEECH_INPUT: {
         if (resultCode == RESULT_OK && null != data) {

            ArrayList<String> result = data
                  .getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
            System.out.println(result.get(0));

            String ans = "";
            if(result.get(0).equalsIgnoreCase("option A")){
               ans = "a";
               rec = true ;
            }

            if(result.get(0).equalsIgnoreCase("option B")){
               ans = "b";
               rec = true ;
            }

            if(result.get(0).equalsIgnoreCase("option C")){
               ans = "c";
               rec = true ;
            }
```

```
                if(result.get(0).equalsIgnoreCase("option D")){
                    ans = "d";
                    rec = true ;
                }

                if (rec) {
                    if(ans.equalsIgnoreCase(answer))
                        no_correct_answers++;
                        speakQuestion();
                }
                else{
                    q--;
                    t1.speak("Answer not recognised", TextToSpeech.QUEUE_FLUSH, null);
                    speakQuestion();
                }




            }
            break;
        }

    }
  }


}

package com.mirror.blindquizv2;

/**
 * Created by arunkumar on 4/4/18.
```

```
 */

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class BQExamHomeActivity extends Activity implements View.OnClickListener {


    Button mbutton_submit;
    EditText medit_subject;

    String tot;
    String correct;



    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.examhome);


        mbutton_submit = (Button) findViewById(R.id.button_submit);
        medit_subject = (EditText) findViewById(R.id.edit_subject);


        mbutton_submit.setOnClickListener(this);
```

```
    }

    public void onClick(View v) {
        // TODO Auto-generated method stub

        switch (v.getId()){

            case R.id.button_submit:

                String subject = medit_subject.getText().toString();
                System.out.println("Subject Name : " + subject);

                Bundle extras = new Bundle();
                extras.putString("subject",subject);

                Intent i = new Intent(this, BQExamActivity.class);
                i.putExtras(extras);
                startActivity(i);
                break;


        }




    }


}
package com.mirror.blindquizv2;

import android.os.Bundle;
import android.speech.tts.TextToSpeech;
```

```
import android.util.Log;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import java.util.Locale;

public class BQMainActivity extends AppCompatActivity {

    TextView mscore;

    String tot;
    String correct;

    TextToSpeech t1;



    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        Bundle extras = getIntent().getExtras();
        tot = (String) extras.getString("total");
        correct = (String) extras.getString("correct");



        mscore = (TextView) findViewById(R.id.text_score);
        mscore.setText("You have answered " + correct + " out of " + tot + "questions");
```

```
        t1=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
               if (status == TextToSpeech.SUCCESS) {

                  int result = t1.setLanguage(Locale.US);

                  if (result == TextToSpeech.LANG_MISSING_DATA
                        || result == TextToSpeech.LANG_NOT_SUPPORTED) {
                     Log.e("TTS", "This Language is not supported");
                  } else {
                     //btnSpeak.setEnabled(true);
                    // speakOut();
                     t1.speak("You have completed your exam successfully.\n You have answered
" + correct + " out of " + tot + "questions", TextToSpeech.QUEUE_FLUSH, null);
                  }

               } else {
                  Log.e("TTS", "Initilization Failed!");
               }
            }

        });









    }
```

```
}
package com.mirror.blindquizv2;

import android.app.Activity;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.DefaultRetryPolicy;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.RetryPolicy;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

public class BQSetQuestionActivity extends Activity implements OnClickListener {
    /**
     * Called when the activity is first created.
     */

    private EditText medit_subject;
    private EditText medit_qno;
    private EditText medit_question;
```

```
private EditText medit_optiona;
private EditText medit_optionb;

private EditText medit_optionc;
private EditText medit_optiond;
private EditText medit_answer;

private Button mbutton_submit;

String subject;
String qno;
String question;
String optiona;
String optionb;
String optionc;
String optiond;
String answer;

ProgressDialog progressDialog;



Dialog dialog;
String ip;

@Override
public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.setquestion);


   medit_subject = (EditText) findViewById(R.id.edit_subject);
   medit_qno = (EditText) findViewById(R.id.edit_qno);
   medit_question = (EditText) findViewById(R.id.edit_question);
```

```
        medit_optiona = (EditText) findViewById(R.id.edit_optiona);
        medit_optionb = (EditText) findViewById(R.id.edit_optionb);
        medit_optionc = (EditText) findViewById(R.id.edit_optionc);
        medit_optiond = (EditText) findViewById(R.id.edit_optiond);
        medit_answer = (EditText) findViewById(R.id.edit_answer);

        mbutton_submit = (Button) findViewById(R.id.button_submit);

        mbutton_submit.setOnClickListener(this);



    }

    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub

        setQuestion();
    /*     subject = medit_subject.getText().toString();
        qno = medit_qno.getText().toString();
        question = medit_question.getText().toString();
        optiona = medit_optiona.getText().toString();
        optionb = medit_optionb.getText().toString();
        optionc = medit_optionc.getText().toString();
        optiond = medit_optiond.getText().toString();
        answer = medit_answer.getText().toString();
        //   Bundle bundle = new Bundle();
        //   bundle.putString("user", user);
        //  bundle.putString("pass", pass);
```

```
        // Instantiate the RequestQueue.
        RequestQueue queue = Volley.newRequestQueue(this);


        String url ="http://mirrortech.in/quiz/register.php?pass=arun" ;
        url = url + "&subject="+subject ;
        url = url + "&question="+question ;
        url = url + "&qno="+qno;
        url = url + "&optiona="+optiona ;
        url = url + "&optionb="+optionb ;
        url = url + "&optionc="+optionc ;
        url = url + "&optiond="+optiond ;
        url = url + "&answer="+answer ;
        url = url + "&type=add" ;


        System.out.println("URL : " + url);
        url = url.replaceAll(" ", "%20");

// Request a string response from the provided URL.
        StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        // Display the first 500 characters of the response string.
                        // mTextView.setText("Response is: "+ response.substring(0,500));


                        Toast.makeText(BQSetQuestionActivity.this,"Question added successfully.
You can add new question", Toast.LENGTH_SHORT).show();
                        System.out.println("Response :" + response);
                        medit_subject.setText("");
                        medit_qno.setText("");
                        medit_question.setText("");
                        medit_optiona.setText("");
```

```
                medit_optionb.setText("");
                medit_optionc.setText("");
                medit_optiond.setText("");
                medit_answer.setText("");
            }
        }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            System.out.println("Error in getting response");
            error.printStackTrace();
        }
    });

// Add the request to the RequestQueue.
    queue.add(stringRequest);



*/
   }

   public void setQuestion(){

      RequestQueue queue = Volley.newRequestQueue(this);
    /*   String mobile_no = mmobile.getText().toString();
      String email_address = memail_address.getText().toString();
      String firstname = mfirstname.getText().toString();
      String latitude = mlatitude.getText().toString();
      String longitude = mlongitude.getText().toString();
      String password = motp.getText().toString(); */

      subject = medit_subject.getText().toString();
      qno = medit_qno.getText().toString();
      question = medit_question.getText().toString();
```

```
optiona = medit_optiona.getText().toString();
optionb = medit_optionb.getText().toString();
optionc = medit_optionc.getText().toString();
optiond = medit_optiond.getText().toString();
answer = medit_answer.getText().toString();

// String lastname = mlastname.getText().toString();

JSONObject req = new JSONObject();
try {
   req.put("subject", subject);
   req.put("qno", qno);
   req.put("question", question);
   req.put("optiona", optiona);
   req.put("optionb", optionb);
   req.put("optionc", optionc);
   req.put("optiond", optiond);
   req.put("answer", answer);
   //  req.put("last_name", lastname);
} catch (JSONException e) {
   // TODO Auto-generated catch block
   e.printStackTrace();
}

String url ="https://d83arun.pythonanywhere.com/addQuestion";

System.out.println("URL :" + url);

JsonObjectRequest jsonobj = new JsonObjectRequest(Request.Method.POST, url,req,
      new Response.Listener<JSONObject>() {
         @Override
         public void onResponse(JSONObject response) {

            Log.d("Chatbot","Success response");
```

```
                System.out.println("Response: " + response);
                progressDialog.dismiss();

try {
if((response.getString("response")).equals("success")){
// sent_otp = response.getString("otp");
// System.out.println("OTP: " + sent_otp);
Toast.makeText(getApplicationContext(),"Question added
successfully",Toast.LENGTH_SHORT).show();
//Toast.makeText(getApplicationContext(),"Login with registered mobile
no",Toast.LENGTH_SHORT).show();
//finish();
medit_subject.setText("");
medit_qno.setText("");
medit_question.setText("");
medit_optiona.setText("");
medit_optionb.setText("");
medit_optionc.setText("");
medit_optiond.setText("");
medit_answer.setText("");


                    // mregister.setVisibility(View.VISIBLE);
                    // mbuttonotp.setVisibility(View.INVISIBLE);
                    //((Activity)getApplicationContext()).finish();
                    //Intent i = new
Intent(getApplicationContext(),RegistrationActivity.class);
                    //startActivity(i);

                }else{
                    Toast.makeText(getApplicationContext(),"Error while Registering
User",Toast.LENGTH_SHORT).show();

                }
```

```
                } catch (JSONException e) {
                    e.printStackTrace();
                }



            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                error.printStackTrace();
                progressDialog.dismiss();
                Toast.makeText(getApplicationContext(),"Error while
connecting",Toast.LENGTH_SHORT).show();

            }
        }
    ){
        //here I want to post data to sever
    };

    System.out.println("In 1");
    // Add the request to the RequestQueue.
    int socketTimeout = 10000;//2 minutes - change to what you want
    //RetryPolicy policy = new DefaultRetryPolicy(socketTimeout,
DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
DefaultRetryPolicy.DEFAULT_BACKOFF_MULT);
    RetryPolicy policy = new DefaultRetryPolicy(socketTimeout, 0,
DefaultRetryPolicy.DEFAULT_BACKOFF_MULT);
    jsonobj.setRetryPolicy(policy);
    queue.add(jsonobj);

    System.out.println("In 2");
```

```
        //initialize the progress dialog and show it
        progressDialog = new ProgressDialog(this);
        progressDialog.setMessage("Adding question");
        progressDialog.show();


    }

}
```

```
package com.mirror.blindquizv2;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class BQUserOptionsActivity extends Activity implements OnClickListener {


    Button mbutton_setquestion;
    Button mbutton_examhome;


    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.user_options);


        mbutton_setquestion = (Button) findViewById(R.id.button_setquestion);
        mbutton_examhome = (Button) findViewById(R.id.button_examhome);
```

```
        mbutton_examhome.setOnClickListener(this);
        mbutton_setquestion.setOnClickListener(this);


    }

    public void onClick(View v) {
        // TODO Auto-generated method stub

        switch (v.getId()){

            case R.id.button_setquestion:

                Intent i = new Intent(this, BQSetQuestionActivity.class);
                startActivity(i);
                break;

            case R.id.button_examhome:

                Intent ii = new Intent(this, BQExamHomeActivity.class);
                startActivity(ii);
                break;

        }




    }


}
package com.mirror.blindquizv2;
```

```
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.navigation.fragment.NavHostFragment;

public class FirstFragment extends Fragment {

    @Override
    public View onCreateView(
            LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState
    ) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_first, container, false);
    }

    public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        view.findViewById(R.id.button_first).setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                NavHostFragment.findNavController(FirstFragment.this)
                        .navigate(R.id.action_FirstFragment_to_SecondFragment);
            }
        });
    }
}
```

```
package com.mirror.blindquizv2;

import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;

public class MainActivityOrg extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_org);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });
    }

    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
package com.mirror.blindquizv2;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.navigation.fragment.NavHostFragment;

public class SecondFragment extends Fragment {
```

```
    @Override
    public View onCreateView(
            LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState
    ) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_second, container, false);
    }

    public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        view.findViewById(R.id.button_second).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                NavHostFragment.findNavController(SecondFragment.this)
                        .navigate(R.id.action_SecondFragment_to_FirstFragment);
            }
        });
    }
}
```

## A.2 SCREENSHOTS

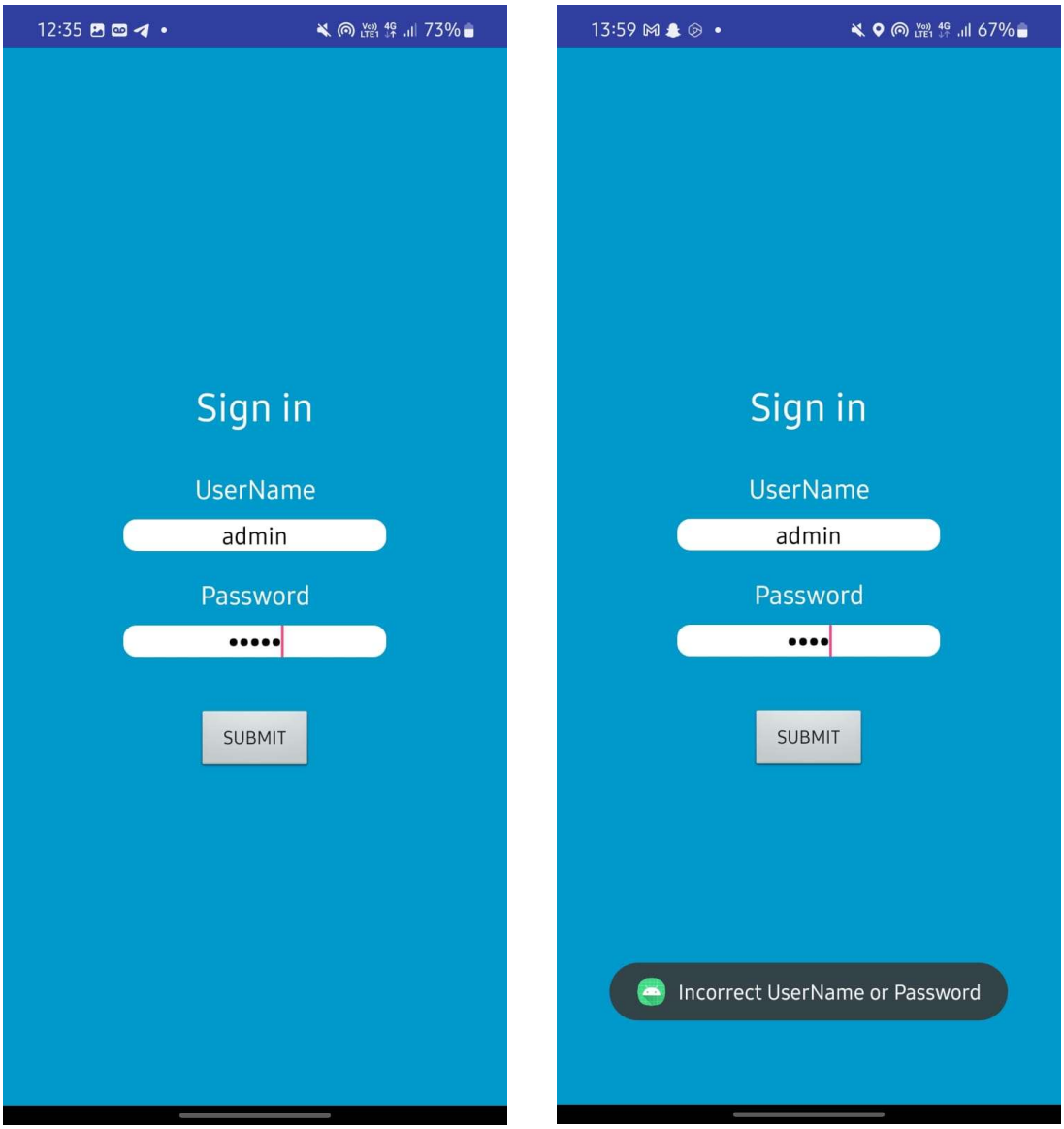



**Fig A.2.1 Login page**

In shown FigA.2.1 Creating a login page in Android Studio involves creating layout XML files for the user interface and implementing the logic in Java or Kotlin files to handle user input and authentication. Below, I'll provide a basic example of how to create a login page in Android Studio: Create a Layout XML Purpose of Login Files Fig A.2.1 res/layout directory in the project explorer. Select "New" > "Layout resource file". Enter a name for the layout file (e.g., activity_login.xml) and click "OK".

In the XML file, define the UI elements for the login page, such as Edit Texts for username and password, and a Button for login. Here's a basic example:

Implement Authentication Logic: In your activity or fragment file (e.g., LoginActivity.java), you'll handle the logic for authentication. Get references to the Edit Texts and Button from the layout XML file. Set an OnClickListener for the login Button to handle user authentication. This could involve validating the username and password entered by the user against a database or server. Here’s a basic example:

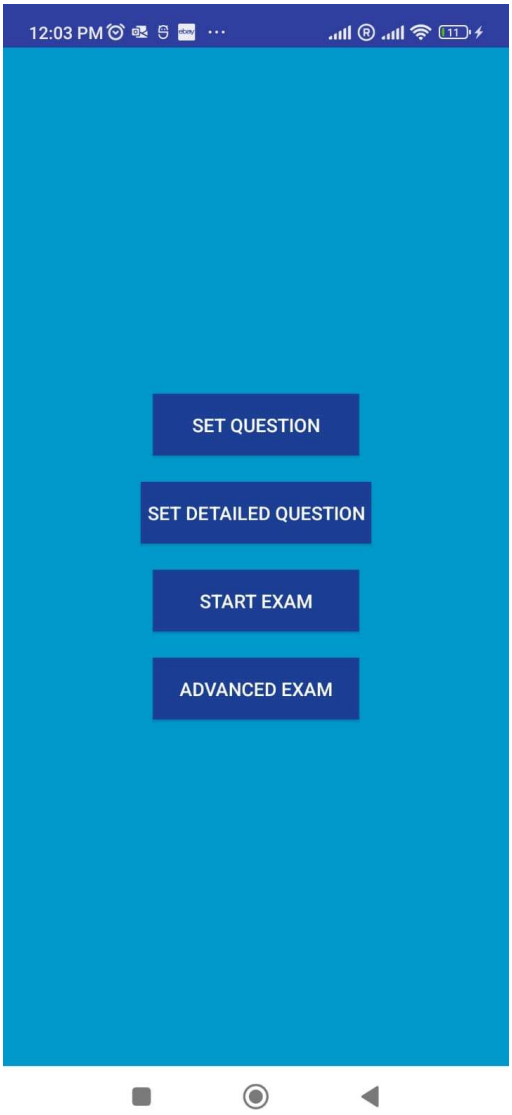



**Fig A.2.2 Set question Are start exam**

For your project, the "Set Question" module serves as a pivotal component designed to facilitate interactive quizzes or assessments within the Android application. This module encompasses various functionalities aimed at seamlessly presenting questions to users, capturing their responses, and providing feedback.

Primarily, the module includes features for loading questions dynamically from a designated data source, such as a database or JSON file, ensuring flexibility in managing question content.

The user interface (UI) is thoughtfully crafted to display questions clearly and intuitively, enabling users to navigate through them effortlessly.

Furthermore, the module incorporates logic for validating user responses, comparing them against correct answers, and calculating scores accordingly. This ensures an engaging and informative experience for users, allowing them to gauge their knowledge or skills effectively.

The "Set Question" module also incorporates elements for tracking user progress, including features to indicate the current question number out of the total set and providing a summary screen at the end of the quiz. Additionally, robust testing procedures are implemented to ensure the reliability and accuracy of the module across various devices and scenarios.

In essence, the "Set Question" module plays a crucial role in enhancing user engagement and interactivity within the application, fostering a dynamic learning or assessment environment tailored to your project's objectives.

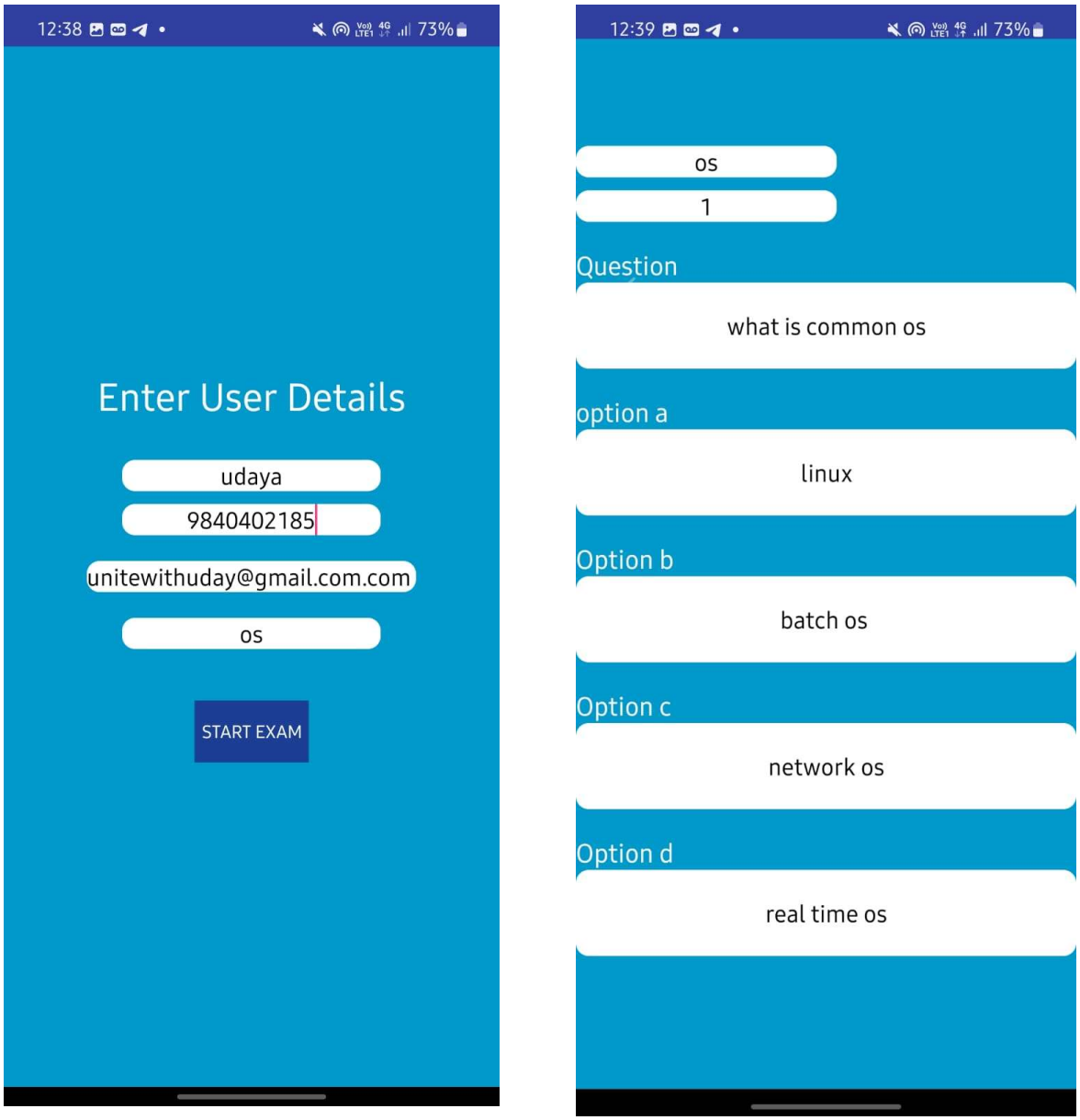



**Fig A.2.3 filling the application and Start exam**

In Fig A.2.3 shows the application is need to be filled for the identification and the Exam is started is the user is ready to take

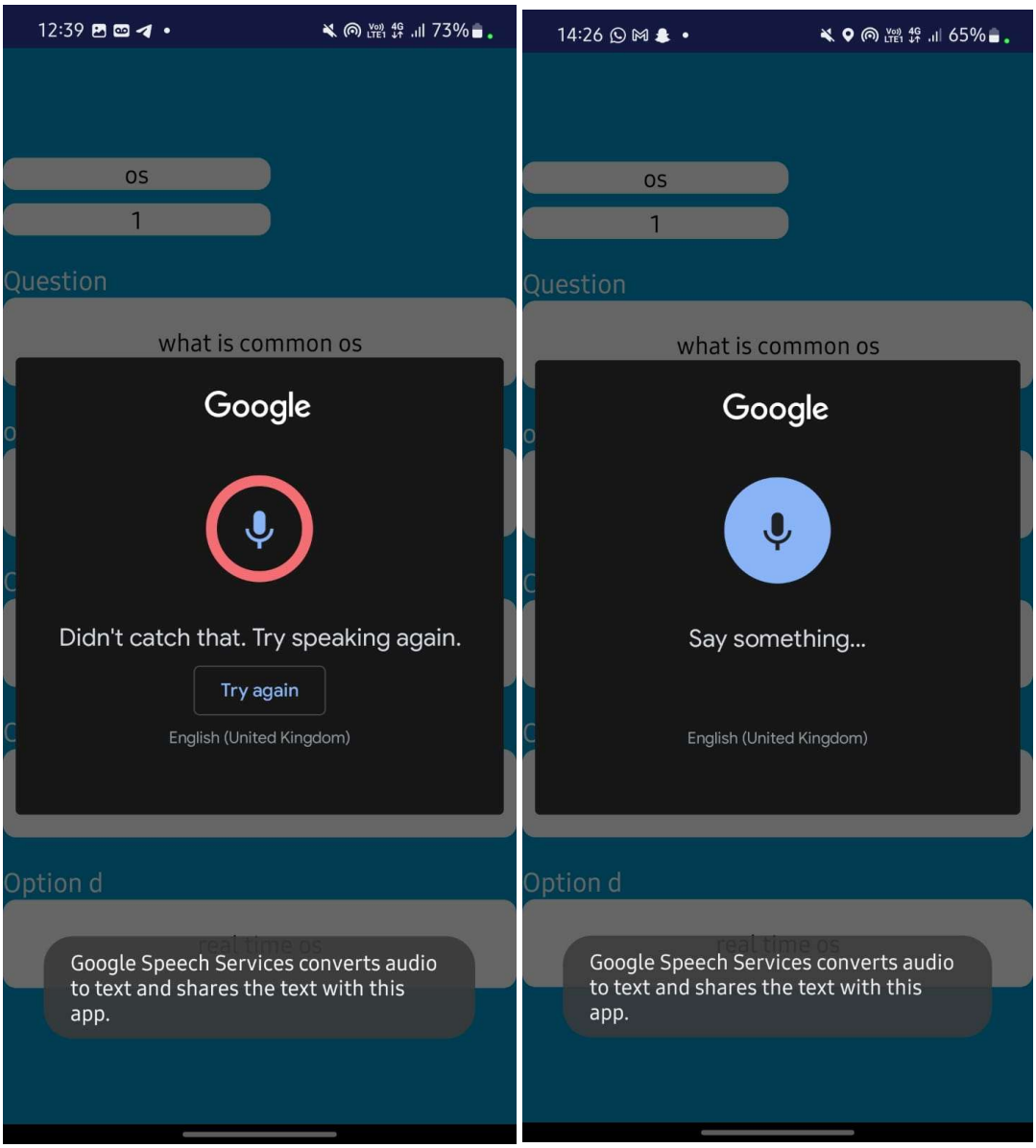



**Fig A.2.4 speech to text synthesis (common for both)**

In the FigA.2.4 shows the "Speech to Text Synthesis" feature serves as a transformative tool that enables users, particularly visually impaired individuals, to interact with the application through spoken commands and receive auditory feedback. This functionality harnesses cutting-edge technology to convert spoken language into written text, thereby enhancing accessibility and inclusivity.

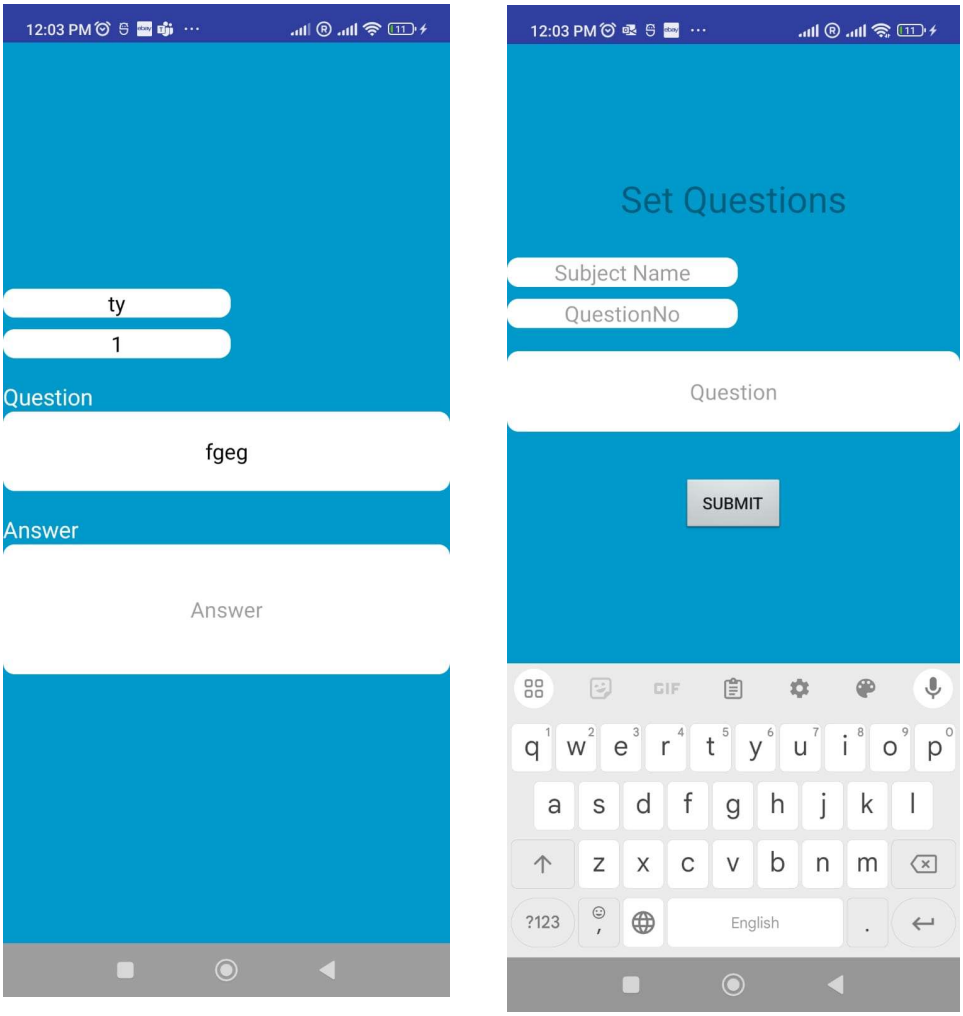



**Fig A.2.6 Advanced test**

In the Fig A.2.6 shows the advanced test provides the paragraph making uses the speech to text synthesis The feature may store past test results for users to reference later, allowing them to track their progress over time and identify trends in their performance.

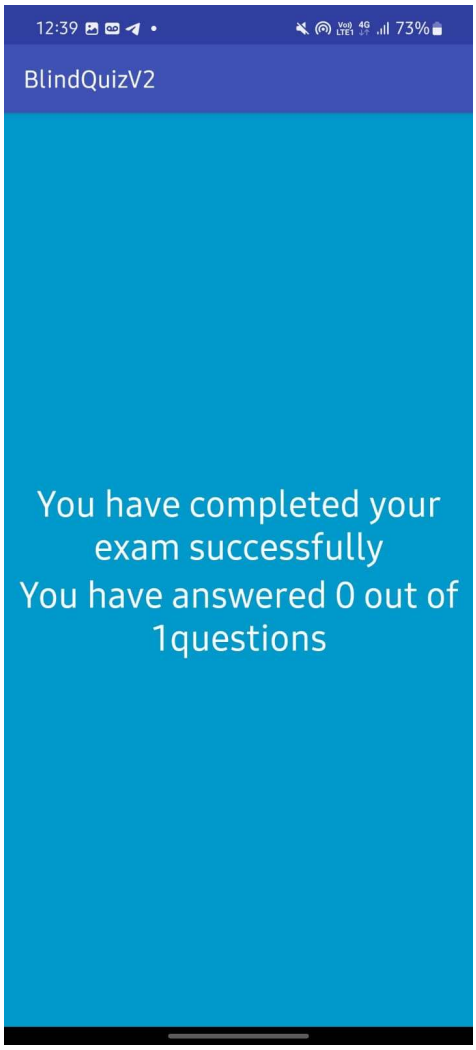



**Fig A.2.6 Final Result page**

In the Fig A.2.6 the "Test Result" feature plays a crucial role in providing users with comprehensive feedback and insights into their performance after completing quizzes or assessments. This feature encompasses various components and functionalities aimed at presenting test results in a clear, informative, and user-friendly manner. Here are some key aspects and functionalities of the "Test

Result" feature:Score Calculation: The feature calculates the user's score based on their responses to the quiz questions. It considers factors such as the number of correct answers, incorrect answers, and any scoring criteria specific to the quiz format.

Visual Representation: The feature may incorporate visual representations, such as charts or graphs, to illustrate the user's performance more intuitively. For example, a bar graph could show the distribution of correct and incorrect answers.

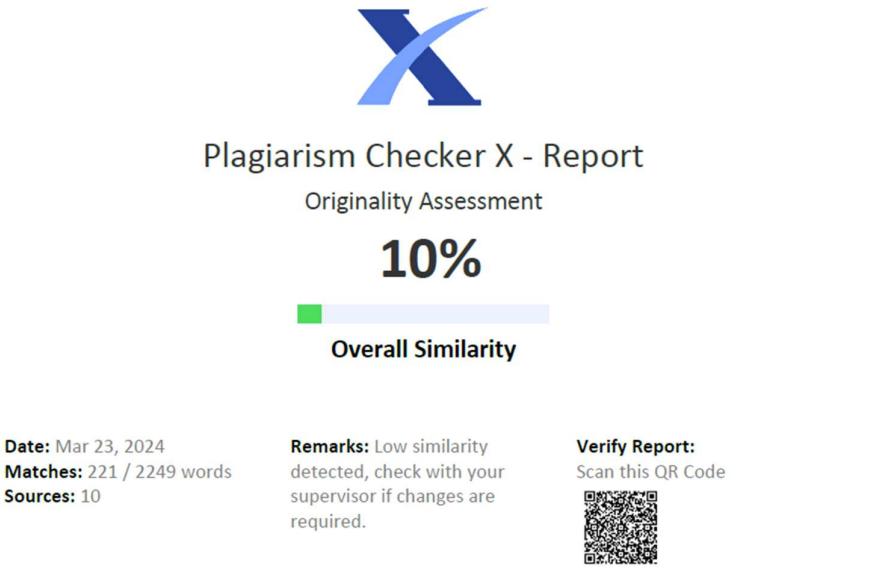
Feedback Messages: The feature may display personalized feedback messages based on the user's performance, encouraging them for their achievements or providing guidance on areas for improvement.

Sharing Options: Users may have the option to share their test results with others, such as teachers, peers, or social networks. This promotes collaboration and facilitates discussion about the quiz content.

Persistence: The feature may store past test results for users to reference later, allowing them to track their progress over time and identify trends in their performance.

Overall, the "Test Result" feature enhances the user experience by providing valuable feedback, insights, and opportunities for reflection, contributing to a more engaging and effective learning or assessment process within your application**.**

## A.3 Plagiarism Report



## A.4 Paper Publication

FORM 2
THE PATENTS ACT 1970(39 of 1970)
&
THE PATENTS RULES,2003 COMPLETE SPECIFICATION
(See section 10 and rule 13)

1.TITLE OF THE INVENTION

**EXAM APP FOR VISUAL IMPAIRED PERSON**

| Applicant | Nationality | Address |
|---|---|---|
| Dr. M.S.VINMATHI | Indian | Department of Computer Science and Engineering, Panimalar Engineering College, Chennai123,Tamilnadu,India |
| DR.N.PUGHAZENDI | Indian | Department of Computer Science and Engineering, Panimalar Engineering College, Chennai123,Tamilnadu,India |
| DHANUSHVARTHAN.RV | Indian | Department of Computer Science and Engineering, Panimalar Engineering College, Chennai123,Tamilnadu,India |
| UDAYAKUMAR .G | Indian | Department of Computer Science and Engineering, Panimalar Engineering College, Chennai123,Tamilnadu,India |
| U.PRASANNAKUMAR | Indian | Department of Computer Science and Engineering, Panimalar Engineering College, Chennai123,Tamilnadu,India |

## FIELD OF INVENTION:

The field of invention for building a system that Exam App for Visual Impaired Person

## BACKGROUND THE INVENTION:

In the landscape of modern education, advancements in technology have been pivotal in facilitating more inclusive learning environments. However, traditional educational systems often fail to adequately accommodate the needs of visually impaired individuals, presenting barriers to accessing educational materials and participating in assessments. Recognizing this challenge, the invention of the inclusive education application emerged as a response to address the accessibility gap and empower visually impaired users in their educational journey.

Visually impaired individuals face significant hurdles when navigating educational resources, particularly in digital formats. Standard user interfaces often rely heavily on visual cues, rendering them inaccessible to those with visual impairments. Moreover, traditional assessment methods such as written exams pose additional challenges, as visually impaired students may require specialized accommodations and assistive technologies to effectively demonstrate their knowledge. These barriers not only hinder academic performance but also perpetuate inequality in educational opportunities.

The inception of the inclusive education application was inspired by the growing demand for innovative solutions that prioritize accessibility and inclusivity in education. Leveraging the capabilities of modern mobile platforms and advancements in voice recognition technology, the application seeks to revolutionize the way visually impaired users interact with educational content. By providing a user-friendly interface with intuitive voice-driven navigation, the application empowers visually impaired individuals to independently access a wide range of educational materials, including textbooks, lectures, and interactive learning modules

.

Central to the invention is the integration of advanced voice recognition technology, which enables seamless interaction with the application's features and functionalities. Through natural language processing algorithms, the application interprets spoken commands and provides real-time auditory feedback, allowing users to navigate menus, select options, and interact with content effortlessly. This innovative approach not only enhances accessibility but also promotes a more engaging and inclusive

learning experience for visually impaired individuals.

Furthermore, the invention harnesses the power of cloud computing to ensure scalability, reliability, and collaborative capabilities. By leveraging cloud-based storage and processing resources, the application can accommodate a diverse range of educational content while maintaining optimal performance across different devices and platforms. Additionally, cloud-based collaboration features enable real-time interaction and communication among users, fostering a supportive learning community.

In summary, the inclusive education application represents a groundbreaking innovation in the realm of educational technology, aiming to break down barriers and empower visually impaired individuals to pursue their academic goals with confidence and independence. By prioritizing accessibility and inclusivity, this invention embodies a transformative approach to education, paving the way for a more equitable and inclusive learning environment for all learners.

**SUMMARY:**

The inclusive education application revolutionizes learning for visually impaired individuals by providing a user-friendly interface powered by advanced voice recognition technology. Addressing the accessibility gap in traditional educational systems, the application allows users to effortlessly navigate educational materials and participate in assessments using spoken commands and auditory feedback. By leveraging cloud computing, the application ensures scalability, reliability, and collaborative capabilities, facilitating a seamless learning experience across different devices and platforms. The integration of voice recognition technology empowers visually impaired users to independently access a wide range of educational resources, promoting inclusivity and equality in education. Through intuitive navigation and real-time interaction features, the application fosters a supportive learning community, enabling users to engage with content and collaborate with peers effectively. Overall, the inclusive education application represents a transformative innovation in educational technology, breaking down barriers and empowering visually impaired individuals to pursue their academic goals with confidence and independence.

**BRIEF DESCRIPTION**:

The inclusive education application is a revolutionary tool designed to empower visually impaired individuals in their educational journey. Built upon the principles of accessibility and inclusivity, this application provides a user-friendly interface that allows users to navigate educational content and

assessments using advanced voice recognition technology.

At its core, the application is designed to break down barriers that hinder visually impaired individuals from accessing educational materials. Through intuitive voice-driven navigation, users can effortlessly explore textbooks, lectures, and interactive learning modules. By simply speaking commands, users can select options, navigate menus, and interact with content in real-time, receiving auditory feedback every step of the way. One of the key features of the application is its seamless integration with cloud computing technology. Leveraging the power of the cloud, the application ensures scalability, reliability, and collaborative capabilities. Users can access educational resources from anywhere, on any device, with the assurance of optimal performance. Additionally, cloud-based collaboration features enable real-time interaction and communication among users, fostering a supportive learning community. The application also includes robust assessment tools, allowing visually impaired individuals to participate in exams and quizzes with ease. Through voice-operated interfaces, users can answer questions and receive immediate feedback, streamlining the assessment process and promoting a more inclusive learning environment. Overall, the inclusive education application represents a transformative innovation in educational technology. By prioritizing accessibility and inclusivity, it empowers visually impaired individuals to pursue their academic goals with confidence and independence. With its intuitive interface, seamless integration with cloud computing, and robust assessment tools, the application aims to break down barriers and promote equality in education for all learners.

**FACTORS INFLUENCING LANDOWNERS' DECISIONS:**

Blind individuals are guided by a multitude of factors when making decisions, with accessibility, independence, social support, and personal preferences playing pivotal roles.Accessibility is a cornerstone factor influencing decision-making for the blind. The availability of tools like assistive technologies and the inclusivity of public spaces heavily impact their ability to engage in daily activities. From employment opportunities to leisure pursuits, the level of accessibility often determines the feasibility of various options.Independence is a deeply ingrained value for many blind individuals, significantly shaping their decisions. The desire for self-reliance influences choices related to mobility, career paths, and living arrangements. Access to training programs and resources that promote independence fosters confidence in decision-making processes. Social support serves as a crucial pillar for blind individuals, offering emotional encouragement and practical assistance. Strong networks comprised of family, friends, and community members provide guidance and reassurance, influencing decisions across different life domains.Personal

preferences and individual values also play a central role in decision-making. Blind individuals, like anyone else, have diverse interests, career aspirations, and lifestyle preferences. These factors guide choices related to education, employment, leisure activities, and relationships.Overall, the decisions made by blind individuals are shaped by a complex interplay of accessibility, independence, social support, and personal preferences. Recognizing and addressing these factors is essential for fostering inclusive environments and empowering blind individuals to make choices that enrich their lives and enable active participation in society.

**WORKING:**

**Android Studio** is a free development suite for Android apps. It's the official integrated development environment (IDE) for Google's Android operating system. It's built on JetBrains' IntelliJ IDEA software and is available for download on Windows, macOS, and Linux based operating systems. Android Studio is a complete IDE that includes a code editor, virtual Android emulator, and code templates. It also contains tools for development, debugging, testing, and performance. Android Studio can run on 4GB RAM, but it lags a bit. 8GB RAM is ideal for running Android Studio as it helps in running their emulators.

**XML** (Extensible Markup Language) is a lightweight markup language used in Android Studio to implement UIrelated data. It's a simple, scalable way to store and organize data. Using this language, the front end of the application is built. The front-end of an application, is the layer or element that the user has the ability to use, see, and interact with through buttons, images, interactive elements, navigational menus, and text

**Java**: Android Studio supports Java 11+ APIs without requiring a minimum API level for your app. This means that if you use an API introduced in Android 13, the code will also work on all previous versions. Some say that Java is the most suitable programming language for developing mobile apps because it allows easy multitasking and offers advanced exception-handling opportunities. Using this language, the back-end of the application is built. Java is a popular choice for building Android app backends. Here are some benefits of using Java for Android app backends: Adaptability: Java's memory management system is versatile and allows for multi-threading.Security: Java is a secure language.Cross-platform use: Java allows for cross-platform use.Developer-friendly: Java is a simple, object-oriented language that's easy to learnFrameworks: Java developers can use frameworks like Spring and Hibernate to build scalable and secure web applications. B.

**Authentication:** This section lets you manage the Firebase Authentication for your project. Using this Firebase Console as a Database for Storage and Authentication, it stores user’s credentials, images and bills given by the user.

**Google console**: Google Search Console is a free service that helps website owners, developers, and SEO professionals understand how their site is performing in Google Search. You can also use the Maps JavaScript API to add a map to your website.This provides imagery and local data from the same source as Google Maps. You can also style the map, visualize your own data, and use services like geocoding and directions. Using this Google console, this application has access to map and finds current location WA business

**Accessibility Simulation:** Design scenarios that represent environments with varying levels of accessibility. For instance, create a model of a workplace with and without assistive technologies like screen readers or tactile signage. Blind individuals can navigate these environments virtually to experience the challenges and advantages each offers.

**Independence Scenarios:** Develop scenarios that highlight the importance of independence in decision-making. For example, simulate situations where blind individuals must choose between relying on assistance and using alternative navigation techniques. Measure the impact on their confidence and autonomy based on the decisions they make.

**Social Support Interactions:** Design interactive scenarios that simulate social interactions and support networks. Participants can engage with virtual characters representing family, friends, and support groups to explore how their input influences decision-making processes. Measure the emotional and practical impact of social support on decisions.

**Personal Preference Exploration**: Create decision-making scenarios that reflect personal interests, aspirations, and values. Participants can choose between various options related to education, employment, leisure activities, and relationships, based on their preferences. Analyze the factors driving their decisions and their satisfaction with the outcomes.

**Integration and Feedback:** Integrate the different components of the working model to reflect the interconnected nature of these factors. Allow participants to navigate through scenarios, make decisions, and receive feedback based on their choices. Collect data on decision-making processes, outcomes, and participant feedback to refine the model.

.

**WE CLAIM:**

**CLAIM 1 Increased Accessibility**: The project aims to increase accessibility for blind individuals by providing them with intuitive tools and interfaces that facilitate easier access to educational materials and resources.

**CLAIM 2 Enhanced Independence**: By integrating advanced technologies and emphasizing skills development, the project seeks to enhance the independence of blind individuals, empowering them to navigate their educational journey with confidence.

**CLAIM 3 Improved Social Inclusion**: Through the creation of a supportive learning community and the incorporation of collaborative features, the project aims to improve social inclusion for blind individuals, fostering a sense of belonging and participation in educational settings.

**CLAIM 4 Streamlined Assessment Processes**: The project endeavors to streamline assessment processes for blind individuals by implementing automated scoring mechanisms and tailored question variations that accommodate diverse needs and preferences.

**CLAIM 5 Promotion of Equal Opportunities**: By prioritizing accessibility and inclusivity, the project aims to promote equal opportunities for blind individuals in education, ensuring that they have the same access to educational resources and opportunities as their sighted peers.

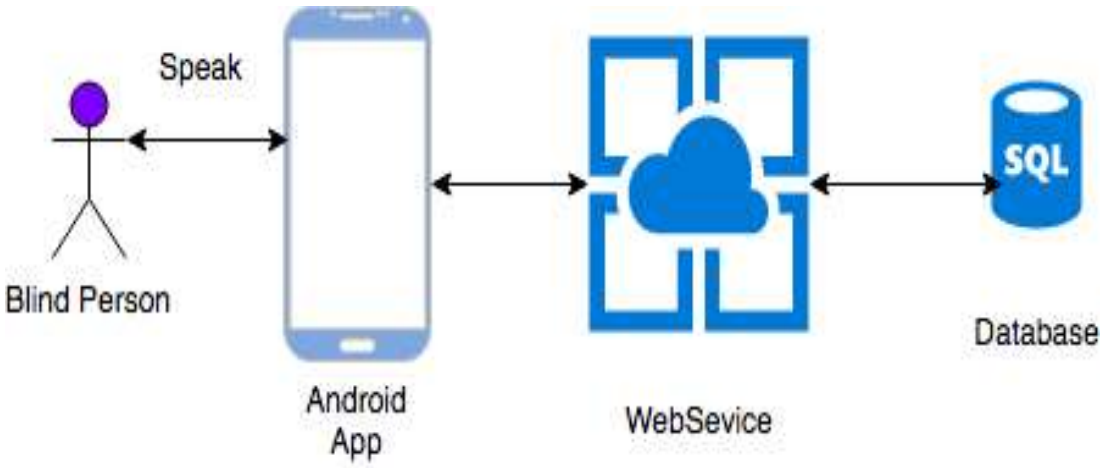
**CLAIM 6 Transformation of Educational Paradigms:** Ultimately, the project seeks to transform educational paradigms by challenging traditional notions of accessibility and fostering a more inclusive learning environment that values the diverse perspectives and abilities of all learners, including those who are blind or visually impaired.

## ABSTRACT:

The Internet, particularly the Web, has revolutionized the way education is delivered, offering a more responsive medium for learners. A web-based examination system accessible via the Internet or intranet facilitates student assessments. Even visually impaired individuals can take online tests using voice-operated Android applications, with questions and choices delivered audibly in real-time. Automated test scoring alleviates faculty of time-consuming tasks, while tailored question variations cater to individual student needs. Tests can be taken at any time and from anywhere, with accessibility options for various users. Results are delivered audibly, and marks are automatically compiled and analyzed for evaluation purposes. In this inclusive environment, visually impaired individuals can access knowledge through colorful interfaces.a

common system designed for visually impaired individuals utilizes voice recognition technology. For instance, on Android phones, there's a feature enabling blind users to interact with apps as seamlessly as sighted users. When they touch any button or navigate through display options, the phone responds immediately with voice feedback. This empowers visually impaired individuals to access all functionalities of the Android phone just like any other user. Keywords: web-based examination system,

**ARCHITECTURE DIAGRAM:**



Welcome Jennifer D Sign out

**Controller General of Patents, Designs & Trade Marks**

सत्यमेव जयते

**G.A.R.6**
**[See Rule 22(1)]**
**RECEIPT**

INTELLECTUAL PROPERTY INDIA
PATENTS | DESIGNS | TRADE MARKS
GEOGRAPHICAL INDICATIONS

**Docket No 45400**

**Date/Time 2024/03/25 13:23:24**

**To**
**Jennifer D**

**UserId: Malar@14**

**Panimalar Engineering College Bangalore**
**Trunk Road, Varadharajapuram,**
**Poonamallee, Chennai- 600123.**

**CBR Detail:**

| Sr. No. | App. Number | Ref. No./Application No. | Amount Paid | C.B.R. No. | Form Name | Remarks |
|---|---|---|---|---|---|---|
| 1 | 202441023306 | TEMP/E-1/27702/2024-CHE | 1600 | 20312 | FORM 1 | EXAM APP FOR VISUAL IMPAIRED PERSON |

| TransactionID | Payment Mode | Challan Identification Number | Amount Paid | Head of A/C No |
|---|---|---|---|---|
| N-0001373022 | Online Bank Transfer | 2503240006012 | 1600.00 | 1475001020000001 |

Total Amount : ₹ 1600.00
Amount in Words: Rupees One Thousand Six Hundred Only

Received from Jennifer D the sum of ₹ 1600.00 on account of Payment of fee for above mentioned Application/Forms.
* This is a computer generated receipt, hence no signature required.

Print

Home About Us Contact Us