A SMART WEB APPLICATION FOR ACCESSIBILITY SUPPORT USING MERN STACK TECHNOLOGY

A Project report submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

Submitted by

G. SRAVAN KUMAR 21Q71A4211

Under the esteemed Guidance of
Mrs. G. RAMA DEVI , M.Tech,
Assistant Professor- AIET
Department of Computer Science and Engineering

Estd. 2005

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(Artificial Intelligence and Machine Learning)

AVANTHI INSTITUTE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

(Approved by A.I.C.T.E., New Delhi, & permanently Affiliated to JNTUGV Vizianagaram Accredited by NAAC With "A+", Grade)
Cherukupally(Village), Near Tagarapuvalasa Bridge,Bhogapuram(Mandal),Vizianagaram(Dist)

(2021 - 2025)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(Artificial Intelligence and Machine Learning)

AVANTHI INSTITUTE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

(Approved by A.I.C.T.E., New Delhi, & permanently Affiliated to JNTUGV Vizianagaram
Accredited by NAAC With "A+", Grade)
Cherukupally(Village), Near Tagarapuvalasa Bridge,Bhogapuram(Mandal),Vizianagaram(Dist)

Estd. 2005

CERTIFICATE

This is to certify that the project report entitled **"A SMART WEB APPLICATION FOR ACCESSIBILITY SUPPORT USING MERN STACK TECHNOLOGY"** submitted by **G. SRAVAN KUMAR (21Q71A4211),** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **CSE-AI&ML** of Avanthi Institute of Engineering and Technology, Vizianagaram is a record of Bonafede work carried out under my guidance and supervision.

(Mrs. G. Rama Devi)
Assistant Professor
Dept. of CSE(AI&ML,DS)
Project Guide

(Mrs. P. Srilakshmi)
HOD – CSE (AI&ML,DS)

(External Examiner)

Acknowledgement

This report would be incomplete without mentioning certain individuals whose guidance and encouragement have been of immense help to complete this Thesis. It is difficult to acknowledge all of them. We can only mention some of them here.

We acknowledge our deep sense of gratitude to our beloved Chairman **Mr. Muttamsetti Srinivasa Rao**, AVANTHI EDUCATIONAL INSTITUTIONS for giving us a chance to do BTech.

We would like to thank to our principal **Dr. B. Murali Krishna,M. Tech,.Ph.D**
Avanthi College of Engineering & Technology for supplying us the best facilities and atmosphere for the project work completion and presentation.

We profusely thank **Mrs. P. Srilakshmi, M.Tech., (Ph.D),** Head of the Department & guidance for encouraging us to keep up this project and for giving prompt suggestions and her valuable guidance, to make this project a success. Her confidence in our programming abilities not only enabled us to work on this project confidently but also instilled courage in us which we had been lacking before we had started this project.

We sincerely express our heartfelt gratitude to **Mrs. G. Rama Devi**, **M.Tech.,** for their invaluable guidance, continuous support, and encouragement throughout the course of this project. Their insightful suggestions and expert advice have played a crucial role in shaping our work and ensuring its success. Their unwavering belief in our capabilities not only motivated us to push our limits but also instilled confidence in us, enabling us to tackle challenges with determination. We are deeply thankful for their time, patience, and mentorship, which have been instrumental in the completion of this project.

TEAM MEMBER

G. SRAVAN KUMAR 21Q71A4211

DECLARATION

We hereby declare that the project work entitled **"A SMART WEB APPLICATION FOR ACCESSIBILITY SUPPORT USING MERN STACK TECHNOLOGY"** submitted to the JNTU-GV is a record of an original work done by under by **G. SRAVAN KUMAR (21Q71A4211),** the esteemed guidance of **Mrs. G. RAMA DEVI , Department of CSE, Computer Science & Engineering,** Avanthi Institute of Engineering & Technology. This project work is submitted in the partial fulfillment of the requirements for the award of the degree **Bachelor of Technology in Computer Science & Engineering – Artificial Intelligence and Machine Learning**. This entire project is done with the best of our knowledge and is not submitted to any university for the award of degree.

(G. SRAVAN KUMAR)
21Q71A4211

Vision and Mission of the Institute

Vision:

To develop highly skilled professionals with ethics and human values

Mission:

1. To impart quality education with industrial exposure and professional training.
2. To produce competent and highly knowledgeable engineers with positive approach.
3. To self confidence among students which is an imperative pre requisite to face the challenges of life.

Computer Science and Engineering
(Artificial Intelligence and Machine Learning)

Vision and Mission of the Department

Vision:

To emerge as recognized center through strong research and teaching environment

to address technological needs of the stakeholders.

Mission:

M1: Provide quality education in both the theoretical and applications of computer science and apply education to solve real-world problems.

M2: Conduct research to advance the state of the art in computer science and integrate research results and innovations into other scientific disciplines.

M3: Graduates with ability to solve complex problems that address societal needs and develop appropriate computer Programs with latest tools and technologies.

M4: Graduates with ability to pursue advanced education, research, other creative and innovative efforts in computer science & engineering for successful career.

Program Educational Objectives (PEOs):

PEO1: Graduates are prepared to apply analysis, predictions, optimization, decision making and develop skills in order to formulate and solve complex problems in the area of Artificial Intelligence and Machine Learning.

PEO2: Graduates are prepared to take up higher studies, research & development and other creative efforts in the area of Artificial Intelligence and Machine Learning which drives scientific and societal advancement through technological innovation and entrepreneurship.

PEO3: Graduates are prepared to use their skills and abilities in an ethical & professional manner.

Program Specific Outcomes(PSOs):

PSO1: Apply Artificial Intelligence and Machine Learning concepts and methods to solve real world problems.

PSO2: Design and develop statistical models for data preparation, exploration, visualization and governance.

PSO3: Adapt to a rapidly changing environment by learning and employing emerging software tools and technologies in the area of Data Science.

Program Outcomes:

PO1: Engineering Knowledge: Apply the knowledge of Mathematics, Science, Engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of Mathematics, Natural Sciences, and Engineering Sciences.

PO3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Remote Accessibility Helper project is designed to empower individuals with disabilities by providing seamless remote assistance and enhancing access to digital and physical environments. By integrating advanced assistive technologies, AI-powered tools, and user-friendly interfaces, this project addresses the unique needs of users with visual, auditory, cognitive, or mobility impairments.

Key features include real-time remote assistance, enabling users to connect with human helpers or AI-based assistants through video, voice, or text-based interactions. The system also offers screen sharing and remote control functionalities for guided support. Additionally, it integrates assistive technologies such as screen readers, braille displays, and voice-controlled interfaces to enhance usability. To improve accessibility, the project incorporates customizable interfaces, including high-contrast modes, adjustable text sizes, and intuitive navigation. Furthermore, it ensures multi-platform support, allowing users to access services across desktops, smartphones, and tablets.

With a strong focus on inclusivity and ease of use, the Remote Accessibility Helper project enhances independence and digital engagement for individuals with disabilities, bridging accessibility gaps through cutting-edge technology.

Contents

CHAPTER - 1
INTRODUCTION

1.1. INTRODUCTION

Accessibility plays a crucial role in ensuring equal opportunities for individuals with disabilities to interact with technology and digital environments. However, many existing systems lack the necessary support to provide seamless accessibility, creating barriers for users with visual, auditory, cognitive, or mobility impairments. The Remote Accessibility Helper project is designed to address these challenges by offering a comprehensive, AI-driven remote assistance system that enhances digital accessibility and real-time support.

This project integrates assistive technologies, AI-powered tools, and user-friendly interfaces to provide personalized support. Key features include real-time remote assistance through video, voice, or text-based interactions, allowing users to seek help from human assistants or AI-based virtual helpers. Additionally, the system supports screen-sharing and remote control functionalities, enabling guided troubleshooting and assistance. The project also ensures compatibility with various assistive devices, including screen readers, braille displays, and voice-controlled interfaces, making it adaptable to diverse user needs.

Developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js), the Remote Accessibility Helper ensures a scalable, flexible, and efficient architecture, allowing seamless real-time communication between users and support systems. Furthermore, its multi-platform support enables accessibility across desktops, smartphones, and tablets, ensuring a wider reach.

By leveraging advanced technology, this project aims to empower individuals with disabilities, enabling them to navigate digital environments effortlessly while fostering independence and inclusivity.

1.2. PROJECT OVERVIEW

The Remote Accessibility Helper is an innovative system designed to enhance digital accessibility for individuals with disabilities, ensuring a seamless and inclusive experience. Developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js), this project integrates real-time remote assistance, AI-powered tools, and interactive features to address various accessibility needs.

Key features include a shopping cart and books-buying section, allowing users to purchase books with ease, and a voice command-based game that provides an engaging and interactive experience.

The system also supports voice-to-text generation and text-to-speech conversion, enabling better communication for individuals with visual or speech impairments. Additionally, screen reader functionality and text size adjustments make navigation smoother and more user-friendly.

To ensure security and personalized accessibility, the project features a robust authentication system with login and sign-up options, where users are assigned a disability ID for verification and customized access settings. With multi-platform compatibility, the system is accessible across desktops, smartphones, and tablets, making it widely available for users.

By combining technology and accessibility, the Remote Accessibility Helper promotes independence, digital inclusivity, and an improved user experience, bridging the gap between technology and accessibility needs.

1.3. PROJECT OBJECTIVE

1. **Developing an Inclusive Accessibility Platform:** The project aims to create a comprehensive system that enhances digital accessibility for individuals with disabilities by integrating assistive technologies and user-friendly features.

2. **Implementing Real-Time Remote Assistance:** Providing video, voice, and text-based support to help users navigate digital and physical environments with ease.

3. **Integrating Speech and Text Accessibility Features :**Enabling voice-to-text conversion and text-to-speech functionality to assist users with communication and information access.

4. **Enhancing User Experience with a Customizable Interface:** Offering screen reader support, text size adjustments, and high-contrast mode to improve navigation for visually impaired users.

5. **Introducing Interactive and Engaging Features:** Developing a voice command-based game for entertainment and cognitive engagement.

6. **Enabling Secure Authentication and Personalization:** Implementing login, sign-up, and disability ID assignment for personalized accessibility settings and secure access.

7. **Providing E-Commerce Accessibility:** Integrating a shopping cart and books-buying section to ensure a seamless online shopping experience.

8. **Ensuring Multi-Platform Support:** Making the system compatible across desktops, smartphones, and tablets for accessibility anytime, anywhere.

9. **Leveraging AI-Powered Assistance:** Using AI-driven chatbots to provide automated responses and real-time support for common user queries.

10. **Promoting Independence and Digital Inclusion:** Empowering users by bridging accessibility gaps, making technology more adaptive, and fostering digital independence.

Overall, the Remote Accessibility Helper aims to provide a comprehensive solution for enhancing digital accessibility, ensuring seamless interaction for individuals with disabilities. By integrating real-time assistance, AI-powered tools, and interactive features, the project focuses on ease of use, quality enhancement, and customization options to cater to diverse user needs.

1.4. PROJECT SCOPE

1. **Develop a Comprehensive Accessibility Platform:** The project aims to build a system that enhances digital accessibility for individuals with disabilities by integrating real-time assistance, AI-powered tools, and adaptive user interfaces to ensure a seamless experience.
2. **Implement Speech and Text-Based Assistive Technologies:** The system will feature voice-to-text and text-to-speech conversion, enabling users with visual or speech impairments to communicate and interact with digital content more effectively.
3. **Provide a User-Friendly and Intuitive Interface:** A well-designed interface will ensure ease of use, offering screen reader support, text size adjustments, high-contrast mode, and simplified navigation for users with different accessibility needs.

4. **Integrate Interactive and Engaging Features:** The project will include a voice command-based game for entertainment and cognitive engagement, promoting inclusivity and interaction for users with mobility impairments.
5. **Enable Secure Authentication and Personalization:** The system will include login, sign-up, and disability ID assignment to ensure secure access and personalized user settings based on accessibility requirements.
6. **Facilitate E-Commerce Accessibility:** A shopping cart and books-buying section will allow users to browse and purchase books conveniently, ensuring an inclusive online shopping experience.
7. **Ensure Multi-Platform Compatibility:** The system will support desktops, smartphones, and tablets, making it widely accessible and adaptable to different devices.
8. **Leverage AI for Real-Time Support:** The platform will integrate AI-driven chatbots and remote assistance tools, enabling users to receive immediate guidance through voice, video, or text-based support.
9. **Support Customization for Enhanced Accessibility:** Users will have the ability to adjust interface settings, modify text sizes, enable screen readers, and choose personalized accessibility preferences, ensuring a tailored experience.
10. **Promote Digital Inclusivity and Independence:** By integrating assistive technologies and user-friendly features, the project will empower individuals with disabilities, promote accessibility in digital environments, and bridge the gap between technology and inclusivity.

OVERALL SCOPE:

The Remote Accessibility Helper project focuses on creating an advanced and inclusive digital accessibility solution by combining real-time assistance, speech-based interaction, AI-powered support, and interactive features. With a strong emphasis on ease of use, security, personalization, and multi-platform accessibility, the project ensures a seamless, independent, and empowering experience for users with diverse needs.

CHAPTER – 2
LITERATURE SURVEY

2. LITERATURE SURVEY

TITLE: "Voice-Controlled Assistive Technology for Individuals with Disabilities"

ABSTRACT: This paper explores the integration of voice-controlled assistive technologies designed to enhance accessibility for individuals with disabilities. It focuses on the development of systems that incorporate speech-to-text and text-to-speech functionalities, enabling seamless communication for visually and speech-impaired users. Advancements in AI-driven screen readers and voice command-based applications are also discussed, highlighting their role in improving interaction with digital content. The research emphasizes the importance of natural language processing (NLP) and deep learning in creating more efficient and user-friendly interfaces. Additionally, the study covers real-time voice recognition, context-aware responses, multilingual support, and personalized voice profiles, which contribute to a more inclusive and intuitive user experience. It also addresses key challenges such as accuracy, latency, and data privacy, proposing adaptive learning algorithms and secure data practices as solutions. Overall, the paper aims to support the development of intelligent, accessible, and inclusive digital environments for individuals with diverse abilities.

TITLE: "AI-Powered Screen Readers and Text Enlargement for Visually Impaired Users"

ABSTRACT: With the rise of digital platforms, accessibility for visually impaired individuals remains a crucial concern. This paper presents an AI-powered screen reader that leverages deep learning techniques to enhance text-to-speech conversion, enabling smoother and more natural voice outputs. The research further investigates text enlargement techniques, contrast adjustments, and real-time text recognition to improve on-screen readability and user experience. Studies suggest that integrating machine learning algorithms with optical character recognition (OCR) can significantly improve text accessibility and user engagement. In addition, the system adapts to user preferences over time by learning interaction patterns, allowing for a personalized and intuitive interface. The implementation of multilingual support ensures accessibility across diverse linguistic backgrounds, while the incorporation of voice commands enables hands-free navigation. This work also addresses challenges such as response time, environmental noise interference, and device compatibility, proposing effective solutions through optimized model training and adaptive design. Ultimately, the paper aims to contribute to the development of inclusive technologies that promote digital equity for visually impaired users.

TITLE: "Enhancing Web Accessibility through Adaptive User Interfaces"

ABSTRACT: The development of adaptive user interfaces designed to enhance web accessibility for users with disabilities. It explores the implementation of customizable settings, including font resizing, color contrast adjustments, screen reader compatibility, and keyboard navigation options. The study evaluates the effectiveness of AI-based recommendations for automatically adjusting accessibility settings based on user preferences, browsing behavior, and contextual cues. By leveraging machine learning algorithms, the system can dynamically adapt the interface in real time, ensuring a seamless and personalized user experience. Experimental results indicate that adaptive interfaces significantly improve usability, engagement, and inclusivity for diverse user groups, including those with visual, motor, and cognitive impairments. Additionally, the paper addresses technical challenges such as cross-

browser support, performance optimization, and compliance with accessibility standards like WCAG (Web Content Accessibility Guidelines). The findings aim to guide developers in creating intelligent, user-centric web platforms that promote digital inclusion and equal access for all users.

TITLE: "Secure Authentication and Disability Identification for Personalized Accessibility"

ABSTRACT: Security and personalization are key aspects of accessibility solutions. This study introduces a disability ID-based authentication system that enables secure user access while providing personalized accessibility settings tailored to individual needs. The research explores advanced multi-factor authentication (MFA) techniques, including biometric verification methods such as facial recognition and fingerprint scanning, to strengthen user identity validation. Additionally, the integration of block chain-based identity management ensures data integrity, transparency, and tamper-proof storage of user credentials. The proposed system not only safeguards sensitive user information but also allows automatic retrieval and application of personalized assistive configurations, such as preferred screen readers, text sizes, color schemes, and navigation modes. Findings suggest that integrating disability ID verification significantly enhances both security and user experience by offering seamless and secure customization of assistive features. The study also discusses challenges related to system scalability, user onboarding, and compliance with data protection regulations, offering practical solutions to address them. Ultimately, this work contributes to the advancement of inclusive, secure, and intelligent accessibility systems.

CHAPTER 3
PROBLEM ANALYSIS

3.1. EXISTING SYSTEM

Individuals with disabilities continue to face significant challenges in accessing digital content due to the lack of built-in accessibility features across many websites and applications. Existing assistive technologies like screen readers and voice recognition tools are often not fully integrated, resulting in a fragmented and inconsistent user experience. Moreover, many of these solutions rely on expensive third-party software or require technical expertise, making them inaccessible to beginners. Key features such as text-to-speech, screen reading, and adaptive UI customization are typically limited in scope and usability. Security is another major concern, as most platforms do not offer secure, disability-friendly authentication methods, making online interactions and account management vulnerable. This highlights the urgent need for a unified, cost-effective, and user-friendly solution that integrates essential features—such as screen reading, voice-to-text conversion, text size adjustments, and JWT-based secure authentication—into one seamless system. Such a comprehensive platform can greatly enhance digital inclusivity, providing a consistent and empowering experience for users with diverse accessibility needs.

3.1.1 Limitation of Existing System

1. **Limited Accessibility Features:** While existing assistive tools provide some support for users with disabilities, they often lack a comprehensive set of features. Many platforms do not offer integrated voice commands, screen reading, or text adjustment tools, making it difficult for users with different needs to interact seamlessly.

2. **Inconsistent User Experience:** Most accessibility tools function as third-party add-ons rather than built-in solutions, leading to inconsistent experiences across different websites and applications. Users may have to rely on multiple tools, each with varying levels of support, reducing overall efficiency.

3. **Customization Constraints:** Existing systems often provide limited customization options for users. While basic features like text resizing and color adjustments may be available, users do not have full control over adapting interfaces to their specific requirements, restricting usability.

4. **Security Concerns:** Many accessibility tools lack strong authentication mechanisms, making them vulnerable to security risks. Without secure login methods like JWT (JSON Web Token), sensitive user data and accounts may be at risk of unauthorized access.

5. **High Resource Consumption:** Some assistive technologies require significant computational resources, leading to performance issues on low-end devices. Features such as real-time speech-to-text conversion and screen reading may consume excessive memory and processing power, affecting usability.

6. **Dependency on External Tools:** Users often rely on expensive or complex third-party software for accessibility enhancements. Free or built-in solutions are either limited in functionality or not widely available, making it difficult for users to find an all-in-one accessibility tool.

3.2. Proposed System

The proposed system, *Remote Accessibility Helper*, is an inclusive platform developed to support individuals with disabilities by integrating multiple accessibility features into a unified and user-friendly environment. It offers essential tools such as a screen reader for visually impaired users, voice-to-text functionality for those with difficulty typing, and text-to-speech capabilities to enhance interaction with digital content. Additionally, the system provides customizable options including text size adjustments, font modifications, and high-contrast modes to improve on-screen readability. An interactive voice command system enables users to navigate, open applications, and interact with web elements through voice control, offering hands-free access especially beneficial for users with mobility impairments. The inclusion of a shopping cart and books-buying section, designed with screen reader compatibility, further ensures that users can independently browse and purchase items with ease.

Security and performance are key components of the system. A secure login and sign-up process is implemented with disability ID-based user verification, while JWT (JSON Web Token) authentication ensures safe and private user sessions. The platform supports real-time text-to-speech and speech-to-text services, allowing users to dictate messages or listen to digital content for better accessibility. It is optimized for smooth performance even on low-end devices and supports a wide range of platforms, including desktops, tablets, and smartphones. By combining advanced assistive technologies with secure authentication and seamless navigation, the Remote Accessibility Helper delivers a comprehensive and scalable solution that enhances digital inclusion and ensures equal access to technology for users with diverse accessibility needs.

3.2.1 Advantages of the Proposed System:

1. **Comprehensive Accessibility Features:** The system includes screen reader support, text-to-speech (TTS), speech-to-text (STT), and adjustable text size, ensuring seamless access to digital content for individuals with disabilities. Voice command support allows users to navigate the platform hands-free, making it more inclusive.

2. **Secure Authentication with JWT:** JSON Web Token (JWT) authentication ensures a secure login process, protecting user data from unauthorized access. Disability ID verification adds an extra layer of security to grant access to relevant accessibility tools.

3. **User-Friendly and Intuitive Interface:** Designed with a simple and easy-to-navigate layout, ensuring a smooth experience for users with varying levels of technical proficiency. Minimal manual input required, thanks to voice-enabled navigation and assistive.

4. **Multi-Functional Platform:** Includes a shopping cart and book-buying section, making e-commerce and educational resources accessible to all users. Allows users to access and purchase products without barriers, improving independence for individuals with disabilities.

5. **Real-Time Assistance and Customization:** Live updates and notifications keep users informed about their activities and important updates. Adjustable text size and contrast settings enhance readability for visually impaired users.

6. **Cost-Effective and Scalable:** Provides an affordable alternative to expensive assistive technologies can be expanded and customized in future versions to accommodate additional accessibility needs.

7. **Cross-Platform Compatibility:** Works efficiently on desktops, tablets, and mobile devices, ensuring accessibility across multiple platforms. Optimized for smooth operation even on low-end devices, making it accessible to a wider user base.

The proposed system not only enhances digital accessibility for individuals with disabilities but also bridges the gap between assistive technology and real-world usability. By integrating essential features such as voice commands, screen reading, text customization, and secure authentication into a single, user-friendly platform, the *Remote Accessibility Helper* ensures a seamless, personalized, and empowering experience. This solution promotes digital inclusion, enabling users to interact confidently with technology across various platforms and everyday scenarios.

CHAPTER – 4
SYSTEM ANALYSIS

4.1 SOFTWARE REQUIREMENTS SPECIFICATION

4.1.1. Non-Functional Requirements:

Non-functional requirements define the quality attributes and constraints that the Remote Accessibility Helper system must adhere to. These requirements focus on aspects such as performance, usability, security, and scalability.

1. **Performance:** The system should efficiently process voice commands, text-to-speech, and speech-to-text operations with minimal latency. It should provide real-time assistance without noticeable delays, ensuring a smooth user experience. The shopping cart and book-buying features should load and respond quickly, even under high user traffic.

2. **Scalability:** The system should be able to handle a growing number of users and requests without significant performance degradation. The authentication process using JWT (JSON Web Token) should be optimized to manage multiple concurrent user sessions securely. The infrastructure should allow easy horizontal scaling to accommodate additional accessibility features in the future.

3. **Usability:** The user interface should be intuitive, user-friendly, and accessible for individuals with disabilities. Features such as screen reader support, text-to-speech (TTS), and adjustable text size should be seamlessly integrated for ease of use. The navigation should be voice-enabled, reducing the need for manual input and making the system accessible for users with mobility impairments.

4. **Reliability:** The system should be highly reliable and stable, ensuring consistent performance across all functionalities. Mechanisms should be in place for error handling and automatic recovery in case of system failures or unexpected crashes. It should ensure persistent user sessions so that progress is not lost during disruptions.

5. **Security:** JWT-based authentication should be implemented to ensure secure user logins and protect sensitive information. All user data should be encrypted to prevent unauthorized access. Security audits should be conducted regularly to identify and mitigate vulnerabilities.

6. **Compatibility:** The system should be compatible with a wide range of devices, including desktops, tablets, and mobile phones. It should support various browsers such as Chrome, Firefox, Edge, and Safari. The system should work efficiently on both Windows and macOS platforms.

7. **Maintenance:** The system should follow a modular architecture to allow easy updates and integration of new accessibility features. It should have logging and monitoring capabilities to track system performance and detect potential issues early. Future updates should be easily deployable without causing downtime or affecting existing user.

4.1.2 Functional Requirements:

User Authentication & Management

- Users must register with a unique username, email address, and password to gain access to the system.
- The system supports JWT-based authentication, ensuring secure login sessions and preventing unauthorized access.
- Users should be able to log in using various methods, including traditional email/password and social media logins like Google or Facebook.
- A password reset and account recovery feature should be available to help users regain access to their accounts if they forget their credentials.

Accessibility Features

- **Screen Reader Support:**

- The system should provide a comprehensive text-to-speech (TTS) functionality that reads aloud text content for visually impaired users.
- Users can easily navigate the system using keyboard shortcuts, which allow them to access key functionalities and content without needing a mouse.

- **Voice-to-Text Generation:**

- Users should be able to dictate their inputs instead of typing, making the system more accessible for individuals with mobility impairments or difficulties with typing.
- The voice recognition feature should support a wide variety of languages and accents to cater to a diverse global user base.

- **Text-to-Speech (TTS) Support:**

- The system should have TTS functionality that reads out text-based content, such as website descriptions, product details, and messages, aloud for users with visual impairments.
- The TTS system should offer customization options, allowing users to adjust voice speed, pitch, and language to better suit their preferences.

- **Text Size Adjustment:**

- Users should be able to easily adjust the size of text to suit their visual needs, including making the text larger or smaller for better readability.
- The system should offer a variety of font styles, including bold or italicized fonts, to enhance legibility.

Shopping Cart & Book Buying Section

- **Product Browsing & Selection:**

- Users should be able to search for products and books using a variety of filters and categories, such as price range, genre, or author.
- Each product or book listing should contain detailed descriptions, high-quality images, and accurate pricing information.

- **Shopping Cart Management:**

- Users should be able to add, remove, and update items in their shopping cart easily, allowing for a smooth shopping experience.
- The system should ensure that the cart retains all user selections even after they log out or close the browser, preventing the loss of items.

Voice Command-Based Game

- **Game Interaction:**

- The system should support full voice control for interacting with games, allowing users to issue commands like "start game," "pause," or "move left."
- Voice input should be mapped to different game actions, such as navigating menus, choosing game modes, and controlling in-game characters.

- **Real-Time Feedback:**

- The system should provide immediate, clear, and understandable audio feedback for every user action, ensuring that users are aware of their progress in the game.
- Users should be able to pause, resume, and restart the game using voice commands, ensuring full accessibility for individuals with physical impairments.

Security & Data Protection

- **JWT-Based Security:**

- The system should use secure JSON Web Tokens (JWT) for authenticating users, ensuring that sessions are encrypted and preventing unauthorized access.
- Each user session should be time-limited, with automatic expiration after a specified period of inactivity, to protect sensitive data.

- **Data Encryption:**

- Sensitive user data, such as personal details and financial information, should be encrypted using industry-standard encryption techniques to prevent unauthorized access.
- The system should periodically update its encryption protocols to comply with evolving security standards.

4.2.Feasibility study

1. Technical Feasibility
2. Operational Feasibility
3. Economic Feasibility

INTRODUCTION

A feasibility study assesses the operational, technical and economic merits of the proposed project. The feasibility study is intended to be a preliminary review of the facts to see if it is worthy of proceeding to the analysis phase. From the systems analyst perspective, the feasibility analysis is the primary tool for recommending whether to proceed to the next phase or to discontinue the project.

The feasibility study is a management-oriented activity. The objective of a feasibility study is to find out if an information system project can be done and to suggest possible alternative solutions.

Projects are initiated for two broad reasons:

1. Problems that lend themselves to systems solutions
2. Opportunities for improving through:

    a. upgrading systems

    b. altering systems

    c. installing new systems

A feasibility study should provide management with enough information to decide:

- Whether the project can be done
- Whether the final product will benefit its intended users and organization
- What are the alternatives among which a solution will be chosen
- Is there a preferred alternative?

TECHNICAL FEASIBILITY

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project. The technical requirements are then compared to the technical capability of the organization. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements.

The analyst must find out whether current technical resources can be upgraded or added to in a manner that fulfils the request under consideration. This is where the expertise of system analysts is beneficial, since using their own experience and their contact with vendors they will be able to answer the question of technical feasibility.

The essential questions that help in testing the operational feasibility of a system include the following:

- Is the project feasible within the limits of current technology?
- Does the technology exist at all?
- Is it available within given resource constraints?
- Is it a practical proposition?
- Manpower- programmers, testers & debuggers
- Software and hardware
- Are the current technical resources sufficient for the new system?
- Can they be upgraded to provide to provide the level of technology necessary for the new system?
- Do we possess the necessary technical expertise, and is the schedule reasonable?
- Can the technology be easily applied to current problems?
- Does the technology have the capacity to handle the solution?
- Do we currently possess the necessary technology?

OPERATIONAL FEASIBILITY

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented.

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the feasibilities to gauge. In order to determine this feasibility, it is important to understand the management commitment to the proposed project. If the request was initiated by management, it is likely that there is management support and the system will be accepted and used. However, it is also important that the employee base will be accepting of the change.

The essential questions that help in testing the operational feasibility of a system include the following:

- Does current mode of operation provide adequate throughput and response time?
- Does current mode provide end users and managers with timely, pertinent, accurate and useful formatted information?
- Does current mode of operation provide cost-effective information services to the business?
- Could there be a reduction in cost and or an increase in benefits?
- Does current mode of operation offer effective controls to protect against fraud and to guarantee accuracy and security of data and information?
- Does current mode of operation make maximum use of available resources, including people, time, and flow of forms?
- Does current mode of operation provide reliable services
- Are the services flexible and expandable?
- Are the current work practices and procedures adequate to support the new system?
- If the system is developed, will it be used?
- Manpower problems
- Labour objections
- Manager resistance
- Organizational conflicts and policies
- Social acceptability
- Government regulations
- Does management support the project?
- Are the users not happy with current business practices?
- Will it reduce the time (operation) considerably?
- Have the users been involved in the planning and development of the project?
- Will the proposed system really benefit the organization?
- Does the overall response increase?
- Will accessibility of information be lost?
- Will the system affect the customers in considerable way?
- Legal aspects
- How do the end-users feel about their role in the new system?
- What end-users or managers may resist or not use the system?
- How will the working environment of the end-user change?

ECONOMIC FEASIBILITY

Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

Possible questions raised in economic analysis are:

- Is the system cost effective?
- Do benefits outweigh costs?
- The cost of doing full system study
- The cost of business employee time
- Estimated cost of hardware
- Estimated cost of software/software development
- Is the project possible, given the resource constraints?
- What are the savings that will result from the system?
- Cost of employees' time for study
- Cost of packaged software/software development
- Selection among alternative financing arrangements (rent/lease/purchase)

The concerned business must be able to see the value of the investment it is pondering before committing to an entire system study. If short-term costs are not overshadowed by long-term gains or produce no immediate reduction in operating costs, then the system is not economically feasible, and the project should not proceed any further. If the expected benefits equal or exceed costs, the system can be judged to be economically feasible. Economic analysis is used for evaluating the effectiveness of the proposed system.

The economic feasibility will review the expected costs to see if they are in-line with the projected budget or if the project has an acceptable return on investment. At this point, the projected costs will only be a rough estimate. The exact costs are not required to determine economic feasibility. It is only required to determine if it is feasible that the project costs will fall within the target budget or return on investment. A rough estimate of the project schedule is required to determine if it would be feasible to complete the systems project within a required timeframe.

4.2. Use Case Scenario

1. **Shopping Cart and Book Buying**

   Actor: User
   Description: A user with accessibility needs wants to purchase books online. They browse the catalog, select books, add them to the shopping cart, and proceed to checkout using accessible navigation options like voice commands or screen readers.

2. **Voice Command-Based Game**

   Actor: User
   Description: A visually impaired user wants to play an interactive game using voice commands. They initiate the game, follow audio cues, and provide voice inputs to control the gameplay.

3. **Voice-to-Text Generation**

   Actor: User
   Description: A student with mobility impairments needs to take notes quickly. They use the system's voice-to-text feature to dictate content, which is automatically transcribed into a digital document.

4. **Text-to-Speech Conversion**

   Actor: User
   Description: A user with reading difficulties uploads a text document and enables text-to-speech conversion. The system reads out the content aloud, allowing the user to understand the material without visual strain.

5. **Screen Reader Support**

   Actor: User
   Description: A visually impaired user navigates through the website using the built-in screen reader support. The system provides audible descriptions of UI elements, allowing the user to interact seamlessly.

6. **Text Size Adjustment**

   Actor: User
   Description: A senior citizen with weak eyesight adjusts the text size of the website for better readability. The system remembers their preference and applies it across all pages.

7. **Authentication and User Verification**
   Actor: System
   Description: The system authenticates users through a secure login process. Users enter credentials, and the system verifies their identity using JWT-based authentication before granting access to accessibility features.

4.3. SYSTEM REQUIREMENTS

4.3.1. HARDWARE REQUIREMENTS

MINIMUM (Required for Execution)		MY SYSTEM (Development)
System	Pentium IV 2.2 GHz	Intel i5 12th gen
Hard Disk	20 GB	500 GB
RAM	1 GB	8 GB

4.3.2. SOFTWARE REQUIREMENTS

Operating System	Windows 10/11
Development Software	Node 22.12.0
Programming Language	Node JS
Integrated Development Environment (IDE)	Visual Studio Code
Front End Technologies	HTML5, CSS3, Java Script, ReactJS
Back End Technologies or Framework	Next JS
Database (NO SQL)	MongoDB
Database Software	MongoDB Server
Web Server or Deployment Server	Nginx Server
Design/Modelling	Rational Rose

CHAPTER -5
SYSTEM DESIGN

5.1. INTRODUCTION

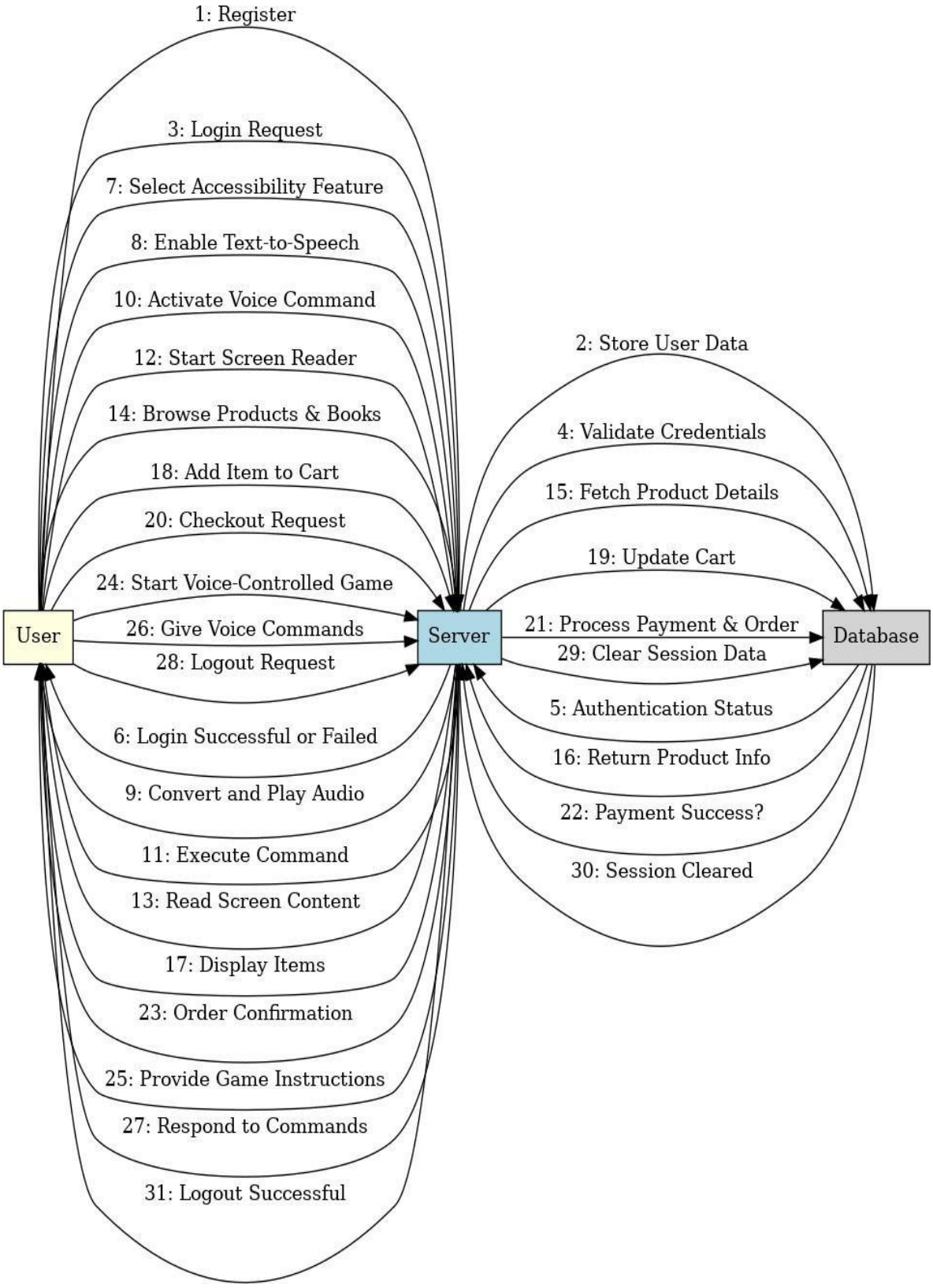
The Remote Accessibility Helper is designed to assist users with disabilities by providing features such as voice command-enabled interactions, screen reading capabilities, and customizable accessibility options. The system integrates various assistive technologies into a cohesive platform, ensuring an intuitive user experience.

With an increasing need for digital inclusivity, individuals with visual impairments, mobility challenges, and other disabilities often face barriers in accessing technology. The Remote Accessibility Helper aims to bridge this gap by offering a user-friendly, accessible platform tailored to meet diverse needs. By incorporating voice-to-text, text-to-speech, screen reading, and other accessibility features, the system ensures that users can navigate digital interfaces effortlessly.

Additionally, this system is designed with security and personalization in mind. Users can customize their accessibility settings, ensuring a tailored experience that meets their specific needs. The inclusion of a shopping cart and book purchasing section further enhances user convenience, allowing seamless access to educational and assistive resources. By leveraging modern technologies such as Node.js, Express.js, MongoDB, and React.js, the platform offers a scalable, efficient, and secure experience.

5.1.1. Sequence Diagram:

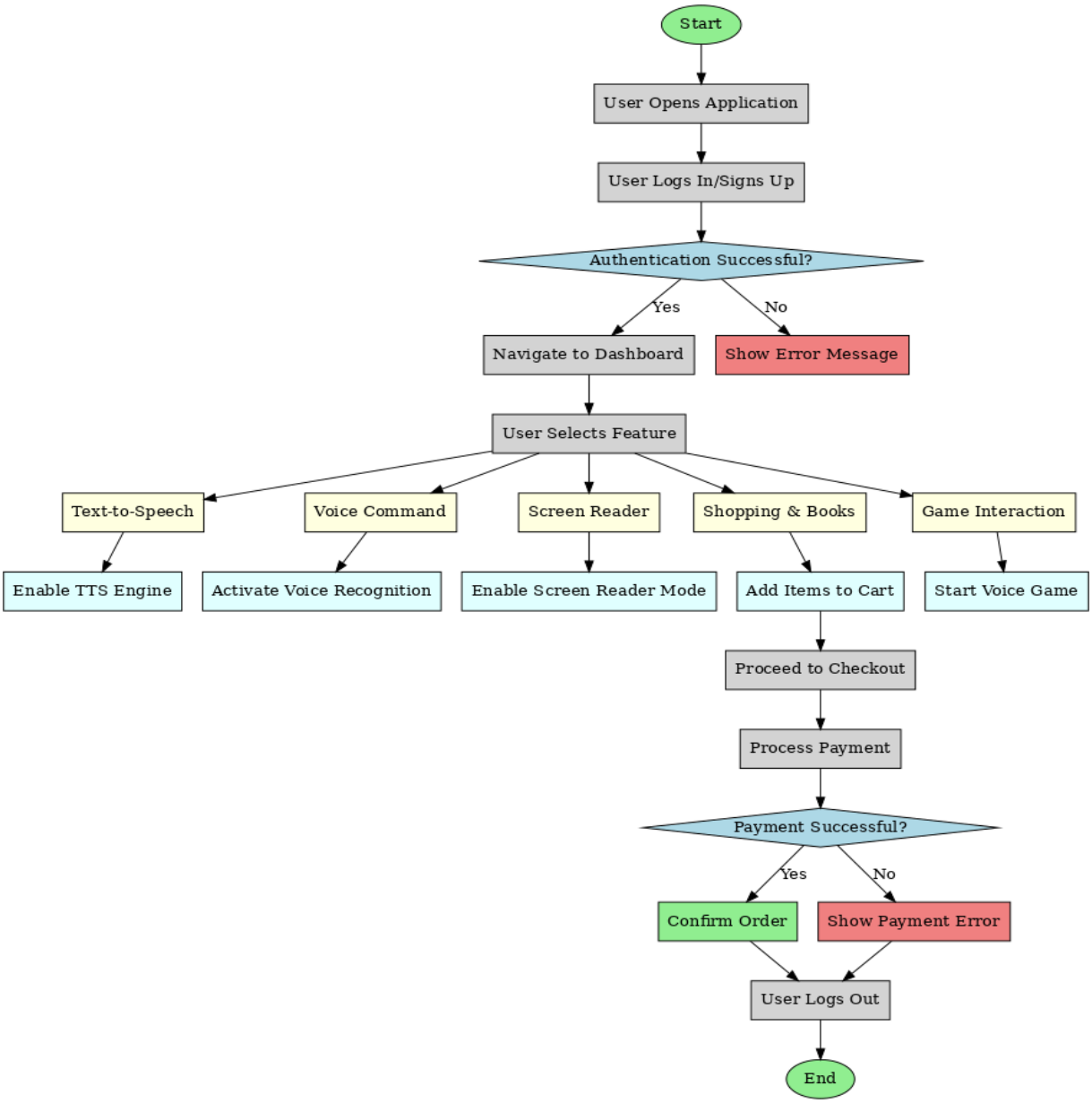
The sequence diagram for the Remote Accessibility Helper project illustrates the interaction between a user and the server. The user can register for an account or log in to an existing account. Once logged in, the user can enable accessibility features such as text-to-speech, voice commands, and screen reading. The server processes these requests and provides the necessary assistance. Additionally, the user can browse and purchase books/products**,** play voice-controlled games**,** and manage their profile. The system ensures secure authentication, real-time updates, and smooth interaction for an enhanced accessibility experience.



Sequence Diagram: User Interaction with Remote Accessibility Helper

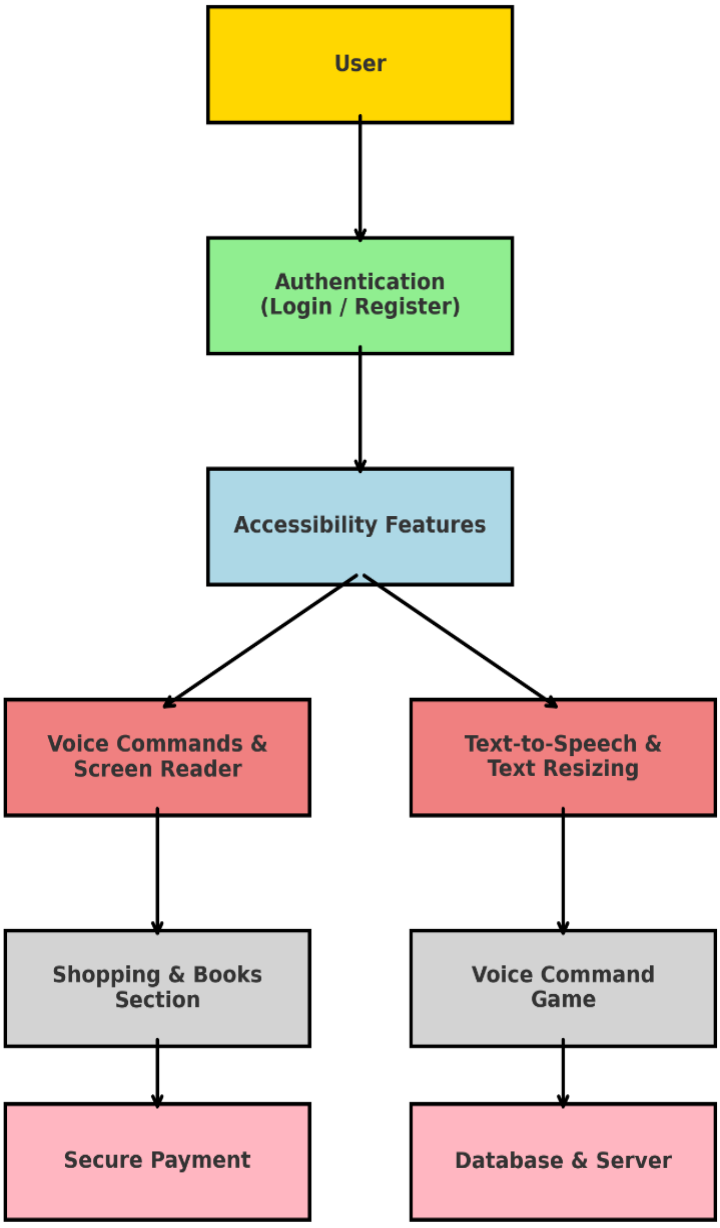
5.1.2 Activity Diagram:

The Remote Accessibility Helper system begins by prompting the user to register or log in. Once authenticated, users can enable accessibility features like text-to-speech, voice commands, screen reading, and text size adjustments. The system processes these requests and enhances user interaction. Users can also browse, purchase books/products, add items to the cart, and proceed with secure payment. Additionally, the system offers a voice-controlled game where users give commands, and the system responds accordingly. After completing activities, users can log out, ensuring secure session termination. This structured workflow provides a seamless, accessible, and user-friendly experience.



Remote Accessibility Helper User Workflow

5.2. SYSTEM ARCHITECTURE



Block Representation of RAH Architecture

5.3. Algorithm Description

The algorithm behind the Remote Accessibility Helper system is designed to deliver seamless, real-time accessibility support through a combination of intelligent modules. At the core lies a user input processing engine that handles both voice and manual inputs. Using speech-to-text APIs and natural language processing (NLP), the system converts spoken commands into structured actions while interpreting user intent through contextual analysis. Keyboard and mouse events are also captured to activate shortcut-based functions like screen reading or contrast toggling. Once input is received, the accessibility feature control module takes over—activating screen readers by parsing the DOM for readable elements and converting them into audio using a text-to-speech (TTS) engine. This engine leverages SSML to modulate voice output, ensuring clarity and natural speech flow.

Simultaneously, the system supports dynamic text resizing and contrast adjustments through JavaScript and CSS variables, allowing users to personalize their reading experience. A voice-to-text module transcribes speech input with real-time grammar correction and multilingual support, enhancing usability for diverse users. For interaction, a navigation engine interprets commands using NLP pipelines and maps them to specific platform actions, such as browsing products, launching a book, or controlling a voice-command-based game. Game control relies on a finite state machine model, ensuring accurate state transitions and feedback.

Security is enforced using JWT (JSON Web Token) authentication, which generates secure tokens during login to validate user identity and protect session integrity. Role-based access control (RBAC) ensures that only authorized users can access sensitive features, with administrative actions logged for transparency. To ensure smooth performance across devices, the system employs lightweight rendering techniques and deferred script loading, optimizing accessibility services for desktops, tablets, and smartphones. Collectively, this algorithmic approach fuses AI, web security, real-time processing, and adaptive UI technologies to empower users and promote inclusive digital experiences.

CHAPTER – 6
IMPLEMENTATION

6.1. Technology Description

Implementation is the stage where the theoretical design is turned in to working system. The most crucial stage is achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after through testing is done and if it found to work according to the specification. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The more complex the system being implemented, more involved will be the systems analysis and design effort required just for implementation. The implementation phase comprises of several activities. The required hardware and software acquisitions are carried out. The System may require some hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

A project is a series of activities that aims at solving particular problems within a given time frame and in a particular location. The activity includes time, money, human and material resources. Before achieving the objectivities, a project goes through several stages. Implementation should take place at and be integrated into all stages of the project cycle. Implementation is the process of having systems personnel check out and put new equipment in to use, train users, install the new application, and construct any files of data needed to it.

6.2. What is Node Programming language?

Node.js is a high-performance, open source and cross platform JavaScript runtime environment that runs on Google's V8 engine. It is designed to build scalable network applications particularly for server-side programming.

1) Event-Driven and Asynchronous

Node.js operates on a non-blocking, event-driven architecture, making it highly efficient for handling multiple simultaneous requests without waiting for previous tasks to complete.

2) JavaScript-Based

Since Node.js runs on JavaScript, developers can use a single language for both front end and backend, making full stack development easier and more seamless.

3) Cross-Platform

Node.js is application can run on Windows, macOS and Linux making it a versatile choice for web application development.

Why Node.js?

Node.js is a high performance which uses the V8 engine to executes JavaScript at high speed.

The Node.js can handles the thousands of concurrent requests efficiently. Because it uses the Libuv along with the V8 engine.

Nodejs uses an event driven architecture, consuming fewer system resources.

Node.js has a vast collection of libraries via npm. Npm is a package manager for Node.js. It was supported by a large community of developers and enterprises.

6.3 Libraries Used

1. **Express.js (Version 4.21.2):**

    - **Description:** A minimal and flexible web application framework for Node.js that provides a robust set of features to develop web and mobile applications.

    - **Use in Project:** Used as the backend framework for handling routes, middleware and API endpoints efficiently.

2. **Mongoose (Version 8.8.4):**

    - **Description:** A MongoDB object modeling tool that provides schema validation, middleware, hooks and easy to use query building.

    - **Use in Project:** Used to interacting with the MongoDB database ensuring schema design and data retrieval operation.

3. **Cloudinary (Version 2.5.1):**

    - **Description:** A cloud-based images and video storage service that offers powerful media management tools

    - **Use in Project:** Used for storing, optimizing and delivering user profiles and course videos.

4. **JSON WEB TOKEN (Version 9.0.2):**

    - **Description:** A compact, URL safe means of representing claims between two parties for secure authentication.

    - **Use in Project:** Used for user authentication and session management, ensuring secure API access and data protection.

6.4. Next JS Framework

Next.js is a powerful React framework that enables developers to build fast, scalable, and SEO-friendly web applications with ease. Built on top of React, Next.js enhances the development experience by offering key features such as server-side rendering (SSR), static site generation (SSG), API routes, automatic code splitting, and optimized performance right out of the box. One of its standout capabilities is hybrid rendering, which allows developers to choose between SSR and SSG for each page based on their needs. Additionally, Next.js provides built-in routing based on the file system, removing the need for complex configuration and making navigation seamless. With built-in image optimization, TypeScript support, and compatibility with popular styling libraries like Tailwind CSS, Next.js is widely adopted for creating production-grade web applications.

In this project, Next.js is utilized as the front-end framework to create a responsive, accessible, and high-performance user interface for the Remote Accessibility Helper. The application structure benefits from Next.js's file-based routing and modular components, which allow efficient page loading and navigation. By leveraging server-side rendering, the platform ensures fast content delivery and better accessibility support for assistive technologies. The integration of Tailwind CSS enables consistent and customizable styling, while React hooks and context APIs manage state and user interaction across components. With its scalability and support for full-stack development (via API routes), Next.js plays a crucial role in delivering an inclusive and seamless digital experience for users with disabilities.

The project follows the Model-View-Controller (MVC) architecture, which separated concerns and improves maintainability:

1. **Model:**
    - Represents the database schema and handles interactions with the MongoDB database using Mongoose.

2. **View:**
    - Build using React.js to provide a dynamic and interactive user interface.
    - Utilizes the Tailwind CSS for styling and React Router for navigation

3. **Controller:**
    - Contains the business logic for handling API requests and responses.
    - Implements authentication, generate questions based on the job description.

6.5. Source code

```
import mongoose from 'mongoose';

const userSchema = new mongoose.Schema({
  username: {
    type: String,
    required: [true, 'Please provide a username'],
    unique: true,
  },
  email: {
    type: String,
    required: [true, 'Please provide an email'],
    unique: true,
  },
  password: {
    type: String,
    required: [true, 'Please provide a password'],
  },
  photo: {
    type: String,
    default:  '/default-avatar.png',
  },
  disabilityId: {
    type: String,
    required: [true, 'Please provide a disability ID'],
    unique: true,
  },
  createdAt: {
    type: Date,
    default: Date.now,
  },
  updatedAt: {
    type: Date,
    default: Date.now,
  },
});

// Update the updatedAt timestamp before saving
userSchema.pre('save', function(next) {
this.updatedAt = Date.now();
  next();
});

export  default  mongoose.models.User  ||  mongoose.model('User',  userSchema);


export function handleKeyboardNavigation(e, actions) {
  switch (e.key) {
    case 'Enter':
```

```
    case ' ':
      if (actions.onActivate) {
        e.preventDefault();
        actions.onActivate();
      }
      break;
    case 'Tab':
      // Let the browser handle tab navigation naturally
      break;
    case 'ArrowUp':
      if (actions.onPrevious) {
        e.preventDefault();
        actions.onPrevious();
      }
      break;
    case 'ArrowDown':
      if (actions.onNext) {
        e.preventDefault();
        actions.onNext();
      }
      break;
    case 'Escape':
      if (actions.onEscape) {
        e.preventDefault();
        actions.onEscape();
      }
      break;
  }
}

// Generate descriptive alt text for images
export function generateAltText(imageName, context = '') {
  // Remove file extension and convert to readable format
  const nameWithoutExtension = imageName.split('.')[0];
  const readableName = nameWithoutExtension
    .replace(/[-_]/g, ' ')
    .replace(/([A-Z])/g, ' $1')
    .toLowerCase();

  return context ? `${context} - ${readableName}` : readableName;
}

// Focus trap for modal dialogs
export function createFocusTrap(containerRef) {
  const focusableElements = containerRef.current?.querySelectorAll(
    'button, [href], input, select, textarea, [tabindex]:not([tabindex="-1"])'
  );

  if (!focusableElements?.length) return;
```

```
  const firstFocusableElement = focusableElements[0];
  const lastFocusableElement = focusableElements[focusableElements.length - 1];

  function handleTabKey(e) {
    if (e.key !== 'Tab') return;

    if (e.shiftKey) {
      if (document.activeElement === firstFocusableElement) {
        e.preventDefault();
        lastFocusableElement.focus();
      }
    } else {
      if (document.activeElement === lastFocusableElement) {
        e.preventDefault();
        firstFocusableElement.focus();
      }
    }
  }

  return handleTabKey;
}

// Announce messages to screen readers
export function announceToScreenReader(message) {
  const announcement = document.createElement('div');
  announcement.setAttribute('aria-live', 'polite');
  announcement.setAttribute('aria-atomic', 'true');
  announcement.setAttribute('class', 'sr-only');
  announcement.textContent = message;

  document.body.appendChild(announcement);

  // Remove the element after a short delay
  setTimeout(() => {
  document.body.removeChild(announcement);
  }, 1000);
}
'use client';

const DEFAULT_AVATAR = '/default-avatar.jpg';
const DEFAULT_PRODUCT_IMAGE = '/product-placeholder.jpg';

import { useState, useEffect } from 'react';
import { useRouter } from 'next/navigation';
import { FaMicrophone, FaVolumeUp, FaVolumeMute } from 'react-icons/fa';

export default function VoiceControls() {
  const router = useRouter();
  const [isListening, setIsListening] = useState(false);
  const [isSpeaking, setIsSpeaking] = useState(false);
```

```
  const [transcript, setTranscript] = useState('');
  const [recognition, setRecognition] = useState(null);
  const [synthesis, setSynthesis] = useState(null);

  useEffect(() => {
    // Initialize speech recognition
    if (typeof window !== 'undefined' && window.SpeechRecognition || window.webkitSpeechRecognition) {
      const SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;
      const recognitionInstance = new SpeechRecognition();
      recognitionInstance.continuous = true;
      recognitionInstance.interimResults = true;

      recognitionInstance.onresult = (event) => {
        const transcript = Array.from(event.results)
          .map(result => result[0])
          .map(result => result.transcript)
          .join('');

        setTranscript(transcript);
        handleVoiceCommand(transcript.toLowerCase());
      };

      setRecognition(recognitionInstance);
    }

    // Initialize speech synthesis
    if (typeof window !== 'undefined' && window.speechSynthesis) {
      setSynthesis(window.speechSynthesis);
    }

    return () => {
      if (recognition) {
        recognition.stop();
      }
      if (synthesis) {
        synthesis.cancel();
      }
    };
  }, []);

  const handleVoiceCommand = (command) => {
    // Navigation commands
    if (command.includes('go to home')) {
      router.push('/');
    } else if (command.includes('go to store')) {
      router.push('/store');
    } else if (command.includes('go to books')) {
      router.push('/books');
```

```
  } else if (command.includes('go to videos')) {
    router.push('/videos');
  } else if (command.includes('go to quiz')) {
    router.push('/quiz');
  } else if (command.includes('go to accessibility')) {
    router.push('/accessibility');
  }

  // Game controls (example commands)
  if (command.includes('start game')) {
    // Add game start logic
    speak('Starting the game');
  } else if (command.includes('pause game')) {
    // Add game pause logic
    speak('Game paused');
  } else if (command.includes('resume game')) {
    // Add game resume logic
    speak('Resuming game');
  }

  // Reading content
  if (command.includes('read page')) {
    const mainContent = document.querySelector('main');
    if (mainContent) {
      speak(mainContent.textContent);
    }
  }
};

const toggleListening = () => {
  if (recognition) {
    if (isListening) {
      recognition.stop();
      setIsListening(false);
      speak('Voice recognition stopped');
    } else {
      recognition.start();
      setIsListening(true);
      speak('Voice recognition started. You can now use voice commands');
    }
  } else {
    alert('Speech recognition is not supported in your browser');
  }
};

const speak = (text) => {
  if (synthesis) {
    synthesis.cancel(); // Stop any current speech
    const utterance = new SpeechSynthesisUtterance(text);
    utterance.onstart = () => setIsSpeaking(true);
```

```
      utterance.onend = () => setIsSpeaking(false);
      synthesis.speak(utterance);
    }
  };

  const stopSpeaking = () => {
    if (synthesis) {
    synthesis.cancel();
    setIsSpeaking(false);
    }
  };

  return (
    <div className="fixed bottom-20 right-4 flex flex-col gap-2" role="region" aria-
label="Voice Controls">
      <button
        onClick={toggleListening}
        className={`p-3 rounded-full ${
          isListening ? 'bg-red-500' : 'bg-blue-500'
        } text-white shadow-lg hover:opacity-90 transition-opacity`}
        aria-label={isListening ? 'Stop voice recognition' : 'Start voice recognition'}
      >
        <FaMicrophone className={isListening ? 'animate-pulse' : ''} />
      </button>

      <button
        onClick={stopSpeaking}
        className={`p-3 rounded-full ${
          isSpeaking ? 'bg-green-500' : 'bg-gray-500'
        } text-white shadow-lg hover:opacity-90 transition-opacity`}
        aria-label={isSpeaking ? 'Stop speaking' : 'Text-to-speech inactive'}
      >
        {isSpeaking ? <FaVolumeUp /> : <FaVolumeMute />}
      </button>

      {transcript && (
        <div className="absolute bottom-full right-0 mb-2 p-2 bg-white rounded shadow-lg
max-w-xs">
          <p  className="text-sm">{transcript}</p>
        </div>
      )}
    </div>
  );
}
```

CHAPTER – 7
TESTING

7.1. INTRODUCTION

Software testing is an investigation conducted to provide clients with information shout the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects)

Testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test have the following Meets the requirements that guided its design and development,

- Responds correctly to all kinds of inputs,
- Performs its functions within an acceptable time,
- ls sufficiently usable,
- Can be installed and run in its intended environments, and holders desire.
- Achieves the general result its stake

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding software bugs (errors or other defects). Software testing can provide objective, independent information about the quality of software and risk of its failure to users and/or sponsors.

Software testing can be conducted as soon as executable software (even if partially complete) exists. The overall approach to software development often determines when and how testing is conducted. For example, in a phased

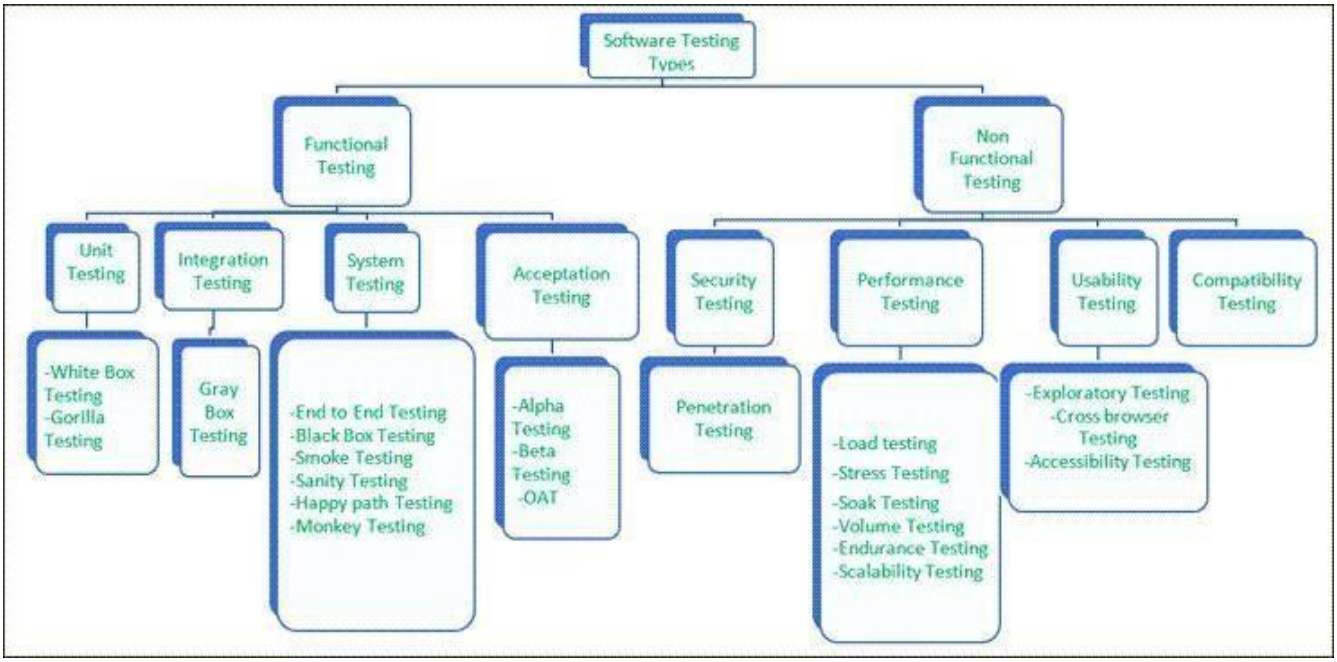
process, most testing occurs after system requirements have been defined and then implemented in testable programs. In contrast, under an Agile approach, requirements, programming, and testing are often done concurrently.

7.2. Types of Software Testing: Different Testing Types with Details

We, as testers, are aware of the various types of Software Testing like Functional Testing, Non-Functional Testing, Automation Testing, Agile Testing, and their sub-types, etc.

Each type of testing has its own features, advantages, and disadvantages as well. However, in this tutorial, we have covered mostly each and every type of software testing which we usually use in our day-to-day testing life.

Different Types of Software Testing



7.2.1. Functional Testing

There are four main types of functional testing.

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

1.Unit Testing:

Unit testing is a type of software testing which is done on an individual unit or component to test its corrections. Typically, Unit testing is done by the developer at the application development phase. Each unit in unit testing can be viewed as a method, function, procedure, or object. Developers often use test automation tools such as NUnit, Xunit, JUnit for the test execution.

Unit testing is important because we can find more defects at the unit test level.

For example, there is a simple calculator application. The developer can write the unit test to check if the user can enter two numbers and get the correct sum for addition functionality.

a). White Box Testing:

White box testing is a test technique in which the internal structure or code of an application is visible and accessible to the tester. In this technique, it is easy to find loopholes in the design of an application or fault in business logic.

b). Gorilla Testing

Gorilla testing is a test technique in which the tester and/or developer test the module of the application thoroughly in all aspects. Gorilla testing is done to check how robust your application is

For example, the tester is testing the pet insurance company's website, which provides the service of buying an insurance policy, tag for the pet, Lifetime membership. The tester can focus on any one module, let's say, the insurance policy module, and test it thoroughly with positive and negative test scenarios.

2). Integration Testing

Integration testing is a type of software testing where two or more modules of an application are logically grouped together and tested as a whole. The focus of this type of testing is to find the defect on interface, communication, and data flow among modules. Top-down or Bottom-up approach is used while integrating modules into the whole system.

This type of testing is done on integrating modules of a system or between systems. For example, a user is buying a flight ticket from any airline website. Users can see flight details and payment information while buying a ticket, but flight details and payment processing are two different systems. Integration testing should be done while integrating of airline website and payment processing system.

a). Gray box Testing

As the name suggests, gray box testing is a combination of white-box testing and black-box testing.Testers have partial knowledge of the internal structure or code of an application.

3). System Testing

System testing is types of testing where tester evaluates the whole system against the specified requirements.

a). End to End Testing

It involves testing a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

For example, a tester is testing a pet insurance website. End to End testing involves testing of buying an insurance policy, LPM, tag, adding another pet, updating credit card information on users' accounts, updating user address information, receiving order confirmation emails and policy documents.

b). Black Box Testing

Black box testing is a software testing technique in which testing is performed without knowing the internal structure, design, or code of a system under test. Testers should focus only on the input and output of test objects.

Detailed information about the advantages, disadvantages, and types of Black Box testing can be found here.

c). Smoke Testing

Smoke testing is performed to verify that basic and critical functionality of the system under test is working fine at a very high level.

Whenever a new build is provided by the development team, then the Software Testing team validates the build and ensures that no major issue exists. The testing team will ensure that the build is stable, and a detailed level of testing will be carried out further.

For example, tester is testing pet insurance website. Buying an insurance policy, adding another pet, providing quotes are all basic and critical functionality of the application. Smoke testing for this website verifies that all these functionalities are working fine before doing any in-depth testing.

d). Sanity Testing

Sanity testing is performed on a system to verify that newly added functionality or bug fixes are working fine. Sanity testing is done on stable build. It is a subset of the regression test.

For example, a tester is testing a pet insurance website. There is a change in the discount for buying a policy for second pet. Then sanity testing is only performed on buying insurance policy module.

e). Happy path Testing

The objective of Happy Path Testing is to test an application successfully on a positive flow. It does not look for negative or error conditions. The focus is only on valid and positive inputs through which the application generates the expected output.

f). Monkey Testing

Monkey Testing is carried out by a tester, assuming that if the monkey uses the application, then how random input and values will be entered by the Monkey without any knowledge or understanding of the application.

4). Acceptance Testing

Acceptance testing is a type of testing where client/business/customer test the software with real time business scenarios.

The client accepts the software only when all the features and functionalities work as expected. This is the last phase of testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

a). Alpha Testing

Alpha testing is a type of acceptance testing performed by the team in an organization to find as many defects as possible before releasing software to customers.

For example, the pet insurance website is under UAT. UAT team will run real-time scenarios like buying an insurance policy, buying annual membership, changing the address, ownership transfer of the pet in a same way the user uses the real website. The team can use test credit card information to process payment-related scenarios.

b). Beta Testing

Beta Testing is a type of software testing which is carried out by the clients/customers. It is performed in the **Real Environment** before releasing the product to the market for the actual end-users.

Beta Testing is carried out to ensure that there are no major failures in the software or product, and it satisfies the business requirements from an end-user perspective. Beta Testing is successful when the customer accepts the software.

Usually, this testing is typically done by the end-users. This is the final testing done before releasing the application for commercial purposes. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area.

So, the end-user uses the software and shares the feedback with the company. The company then takes necessary action before releasing the software worldwide.

c). Operational acceptance testing (OAT)

Operational acceptance testing of the system is performed by operations or system administration staff in the production environment. The purpose of operational acceptance testing is to make sure that the system administrators can keep the system working properly for the users in a real-time environment.

The focus of the OAT is on the following points:

- Testing of backup and restore.
- Installing, uninstalling, upgrading software.
- The recovery process in case of natural disaster.
- User management.
- Maintenance of the software.

7.2.2 Non-Functional Testing

There are four main types of functional testing.

1). Security Testing

It is a type of testing performed by a special team. Any hacking method can penetrate the system.

Security Testing is done to check how the software, application, or website is secure from internal and/or external threats. This testing includes how much software is secure from malicious programs, viruses and how secure & strong the authorization and authentication processes are.
It also checks how software behaves for any hacker's attack & malicious programs and how software is maintained for data security after such a hacker attack.

a). Penetration Testing

Penetration Testing or Pen testing is the type of security testing performed as an authorized cyberattack on the system to find out the weak points of the system in terms of security.

Pen testing is performed by outside contractors, generally known as ethical hackers. That is why it is also known as ethical hacking. Contractors perform different operations like SQL injection, URL manipulation, Privilege Elevation, session expiry, and provide reports to the organization.

Notes: Do not perform the Pen testing on your laptop/computer. Always take written permission to do pen tests.

2). Performance Testing

Performance testing is testing of an application's stability and response time by applying load.

The word stability means the ability of the application to withstand in the presence of load. Response time is how quickly an application is available to users. Performance testing is done with the help of tools. Loader.IO, JMeter, Load Runner, etc. are good tools available in the market.

a). Load testing

Load testing is testing of an application's stability and response time by applying load, which is equal to or less than the designed number of users for an application.

For example, your application handles 100 users at a time with a response time of 3 seconds, then load testing can be done by applying a load of the maximum of 100 or less than 100 users. The goal is to verify that the application is responding within 3 seconds for all the users.

b). Stress Testing

Stress testing is testing an application's stability and response time by applying load, which is more than the designed number of users for an application.

For example, your application handles 1000 users at a time with a response time of 4 seconds, then stress testing can be done by applying a load of more than 1000 users. Test the application with 1100,1200,1300 users and notice the response time. The goal is to verify the stability of an application under stress.

c). Scalability Testing

Scalability testing is testing an application's stability and response time by applying load, which is more than the designed number of users for an application.

For example, your application handles 1000 users at a time with a response time of 2 seconds, then scalability testing can be done by applying a load of more than 1000 users and gradually increasing the number of users to find out where exactly my application is crashing. Let's say my application is giving response time as follows:

- 1000 users -2 sec
- 1400 users -2 sec
- 4000 users -3 sec
- 5000 users -45 sec
- 5150 users- crash – This is the point that needs to identify in scalability testing.

d). Volume testing (flood testing)

Volume testing is testing an application's stability and response time by transferring a large volume of data to the database. Basically, it tests the capacity of the database to handle the data.

e). Endurance Testing (Soak Testing)

Endurance testing is testing an application's stability and response time by applying load continuously for a longer period to verify that the application is working fine.

For example, car companies soak testing to verify that users can drive cars continuously for hours without any problem.

3). Usability Testing

Usability testing is testing an application from the user's perspective to check the look and feel and user-friendliness.

For example, there is a mobile app for stock trading, and a tester is performing usability testing. Testers can check the scenario like if the mobile app is easy to operate with one hand

or not, scroll bar should be vertical, background colour of the app should be black and price of and stock is displayed in red or green colour.
The main idea of usability testing of this kind of app is that as soon as the user opens the app, the user should get a glance at the market.

a). Exploratory testing

Exploratory Testing is informal testing performed by the testing team. The objective of this testing is to explore the application and look for defects that exist in the application. Testers use the knowledge of the business domain to test the application. Test charters are used to guide the exploratory testing.

b). Cross browser testing

Cross browser testing is testing an application on different browsers, operating systems, mobile devices to see look and feel and performance.

Why do we need cross-browser testing? The answer is different users use different operating systems, different browsers, and different mobile devices. The goal of the company is to get a good user experience regardless of those devices.

Browser stack provides all the versions of all the browsers and all mobile devices to test the application. For learning purposes, it is good to take the free trial given by browser stack for a few days.

c). Accessibility Testing

The aim of Accessibility Testing is to determine whether the software or application is accessible for disabled people or not.
Here, disability means deafness, colour blindness, mentally disabled, blind, old age, and other disabled groups. Various checks are performed, such as font size for visually disabled, colour and contrast for colour blindness, etc.

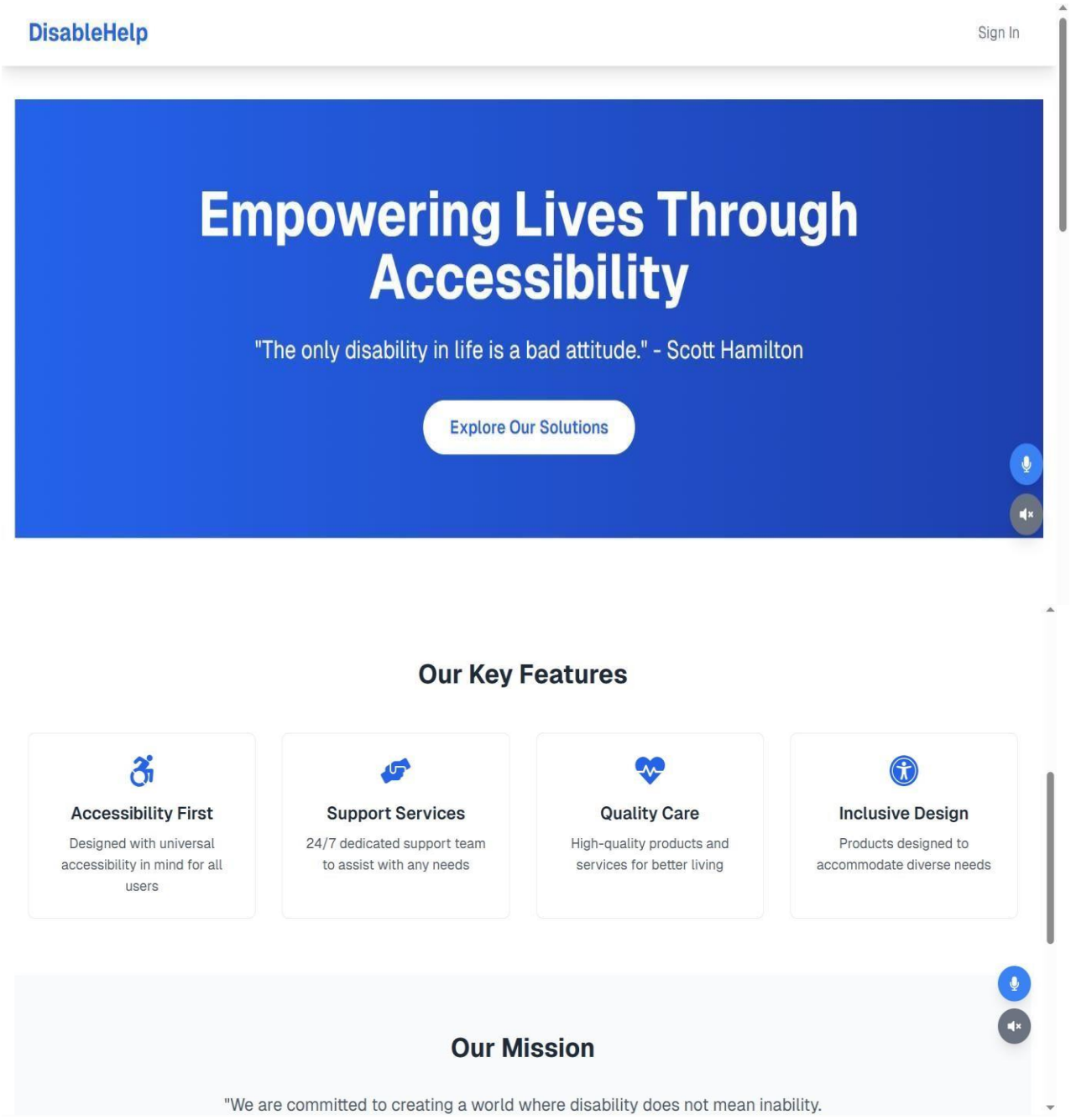
4). Compatibility testing

This is a testing type in which it validates how software behaves and runs in a different environment, web servers, hardware, and network environment.

CHAPTER – 8
RESULTS

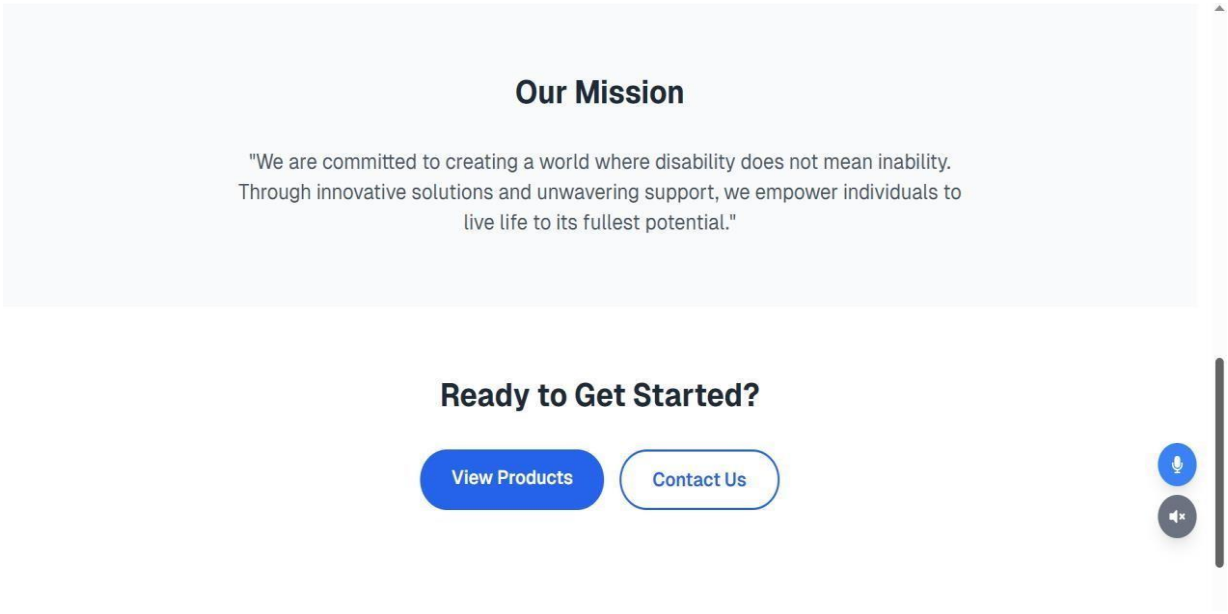
Home Page

The homepage of the *Remote Accessibility Helper*. It features a clean and intuitive navigation bar that includes options like Books**,** Games**,** Login, and Accessibility Tools. The design emphasizes user-friendliness and quick access to core features, supporting seamless navigation for all users, including those with visual or motor impairments



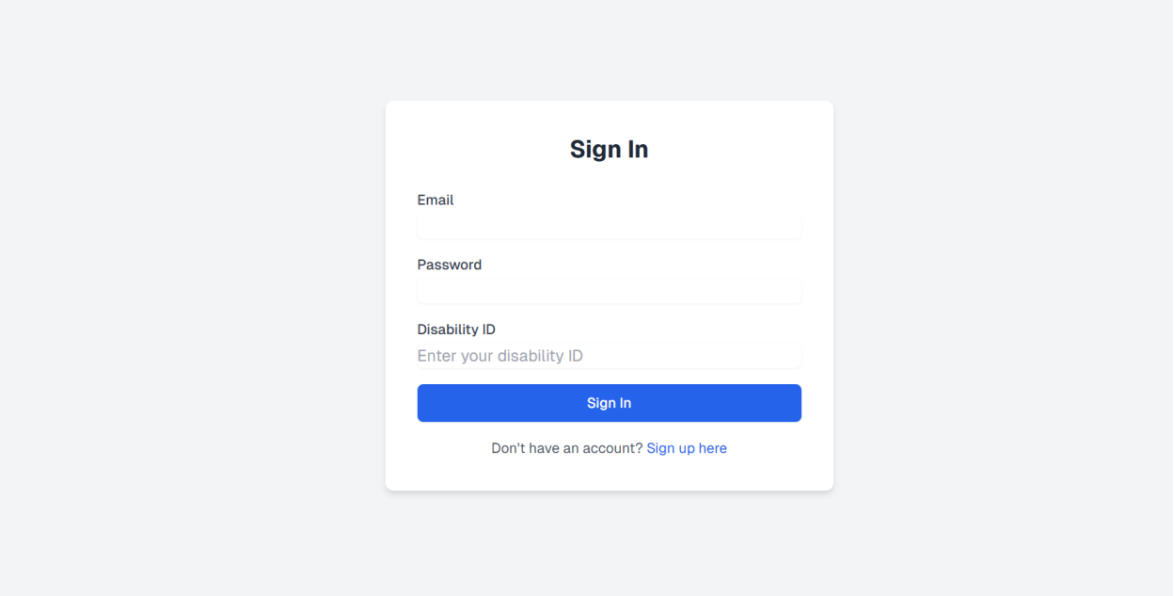
About

Remote Accessibility Helper is a user-friendly web platform designed for individuals with disabilities. It offers voice-controlled games, screen readers, and accessibility tools like text-to-speech and font adjustment. Features like secure login with disability ID and an online bookstore ensure inclusive digital access for all.



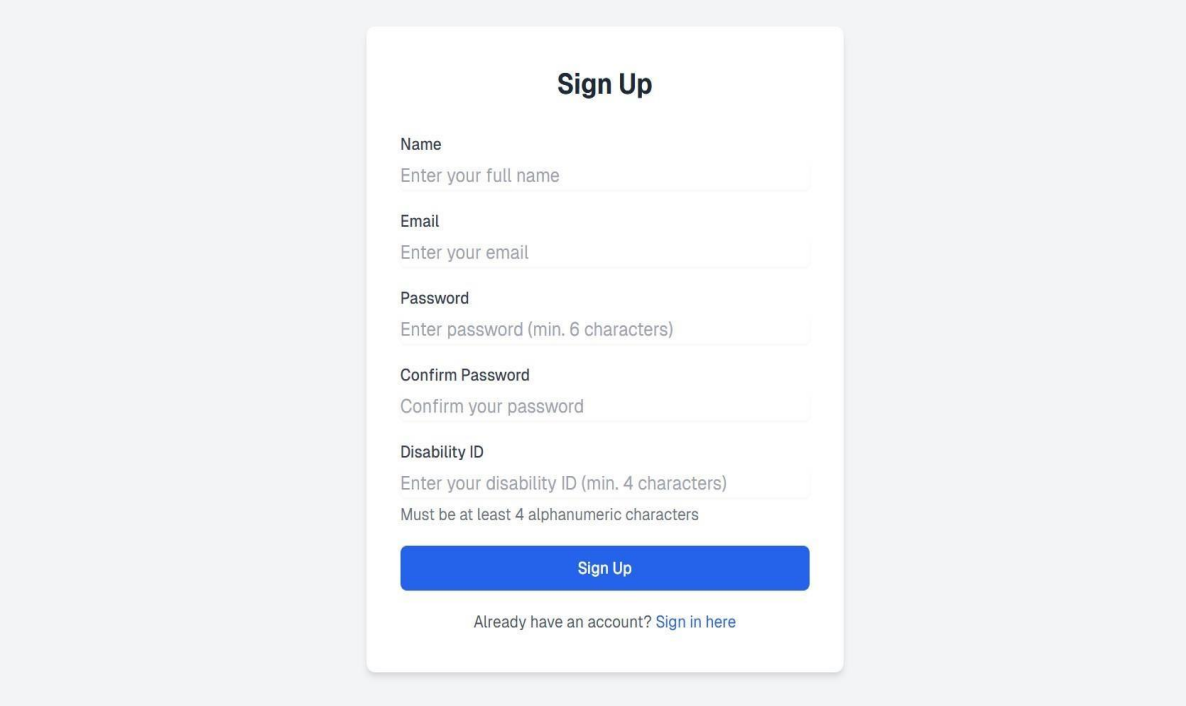
Login Page

The login interface where registered users can securely access the platform. It contains standard fields for email and password, along with buttons for "Login" and navigation to the Sign Up page. The layout ensures accessibility with clearly labeled inputs and responsive elements.

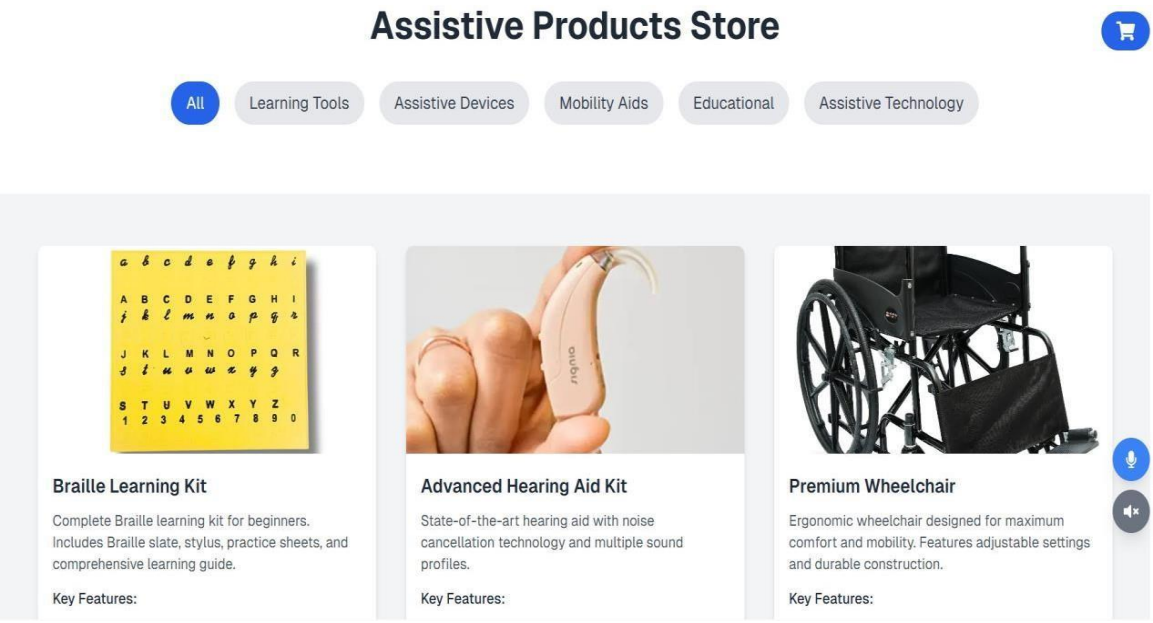


Registration Page

The sign-up screen enables new users to create an account on the platform. The form includes fields such as **Name**, **Email**, **Password**, and **Disability ID**, ensuring authenticated and secure onboarding for users with disabilities. The clean layout and form validation enhance the user experience

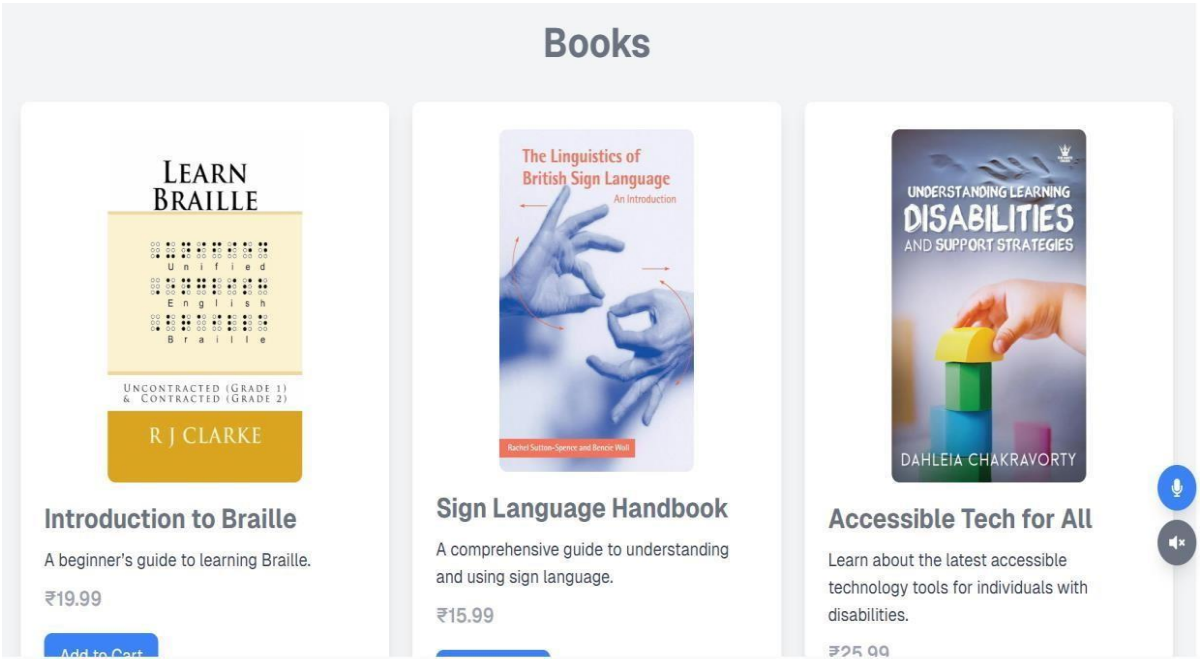


Dashboard

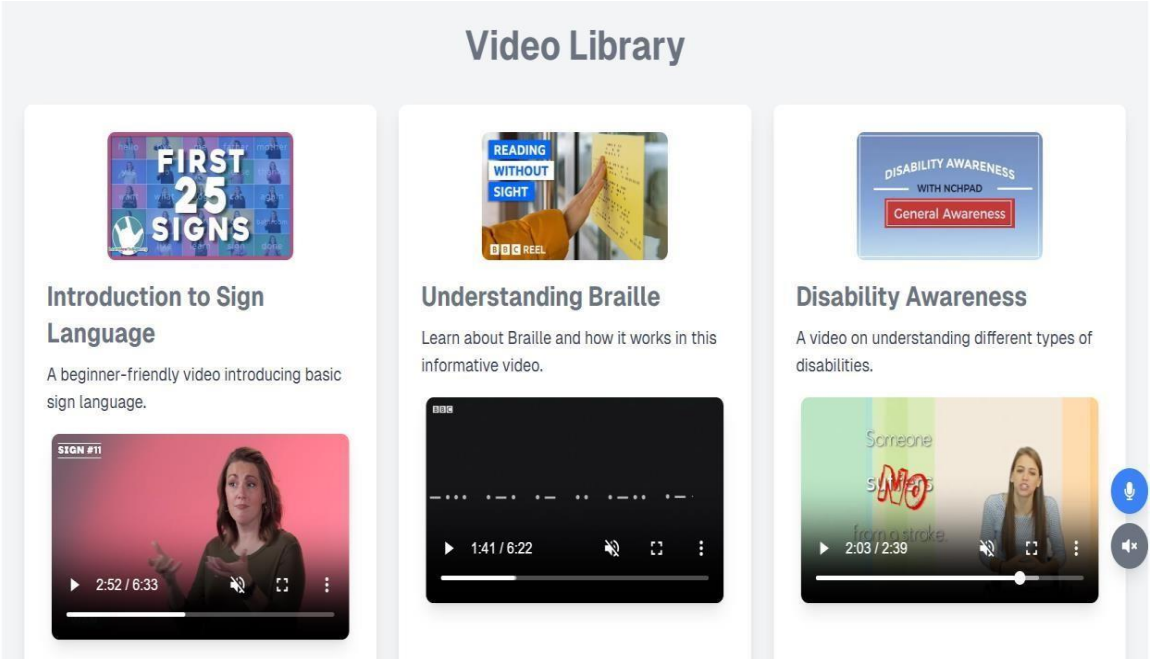


Books

The **book shopping section**, where users can browse various educational and accessible books. Each book is displayed as a card with details and an **"Add to Cart"** button, making it easy for users to select and purchase items. The design promotes inclusivity with large fonts and clear visuals.

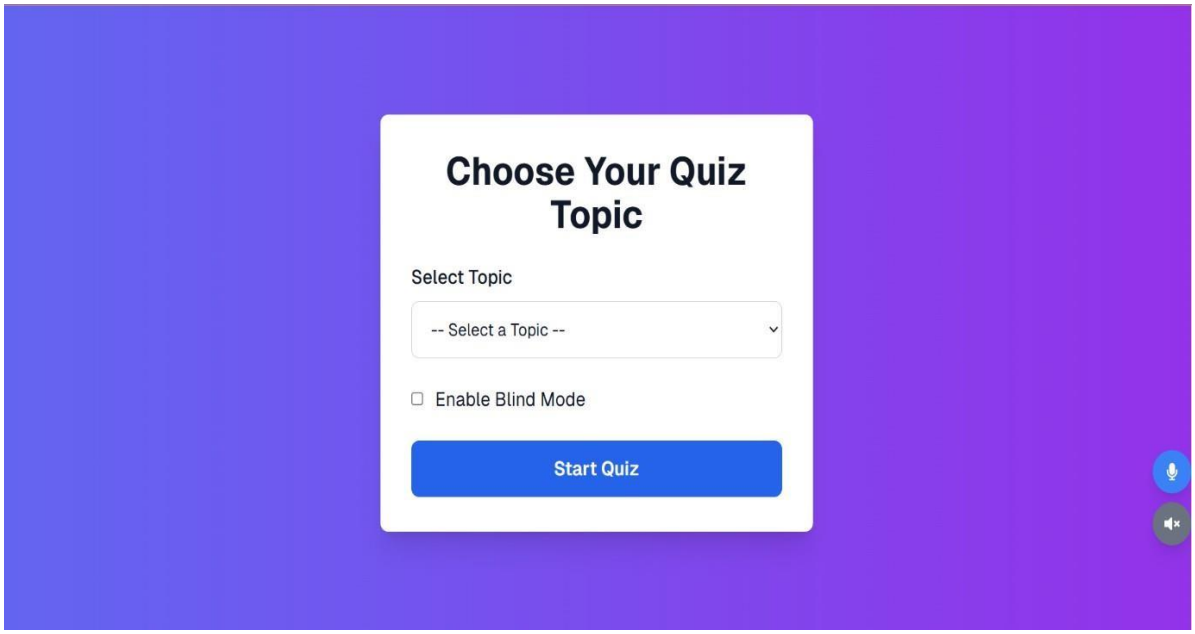


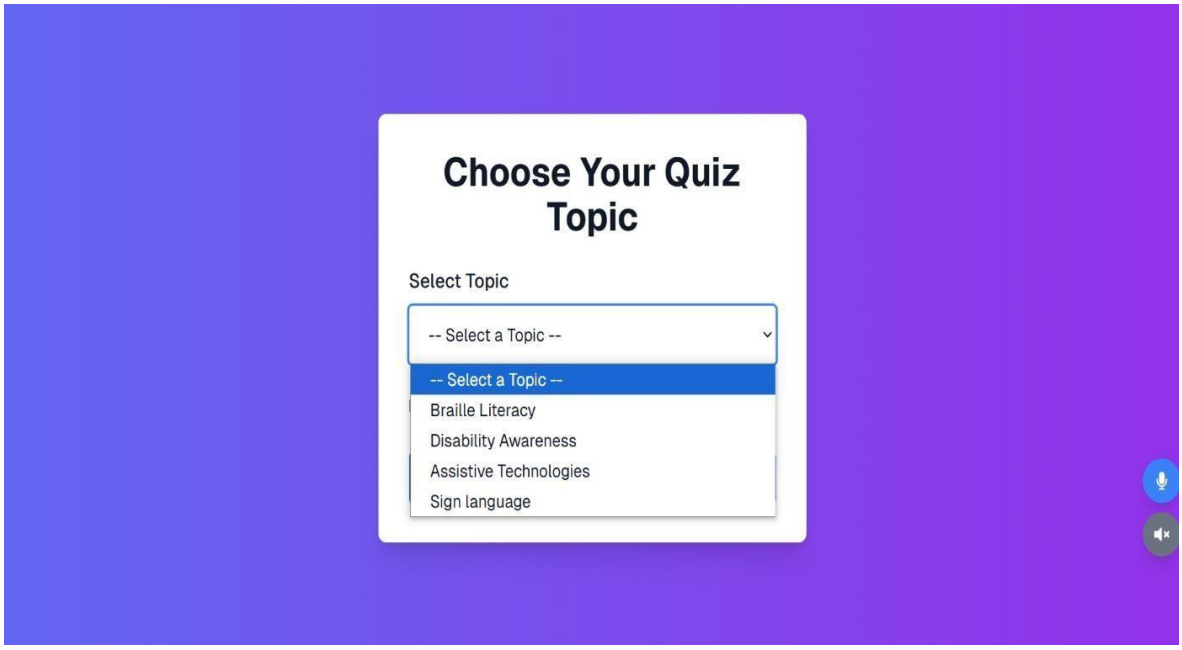
Video Library



Quiz Game

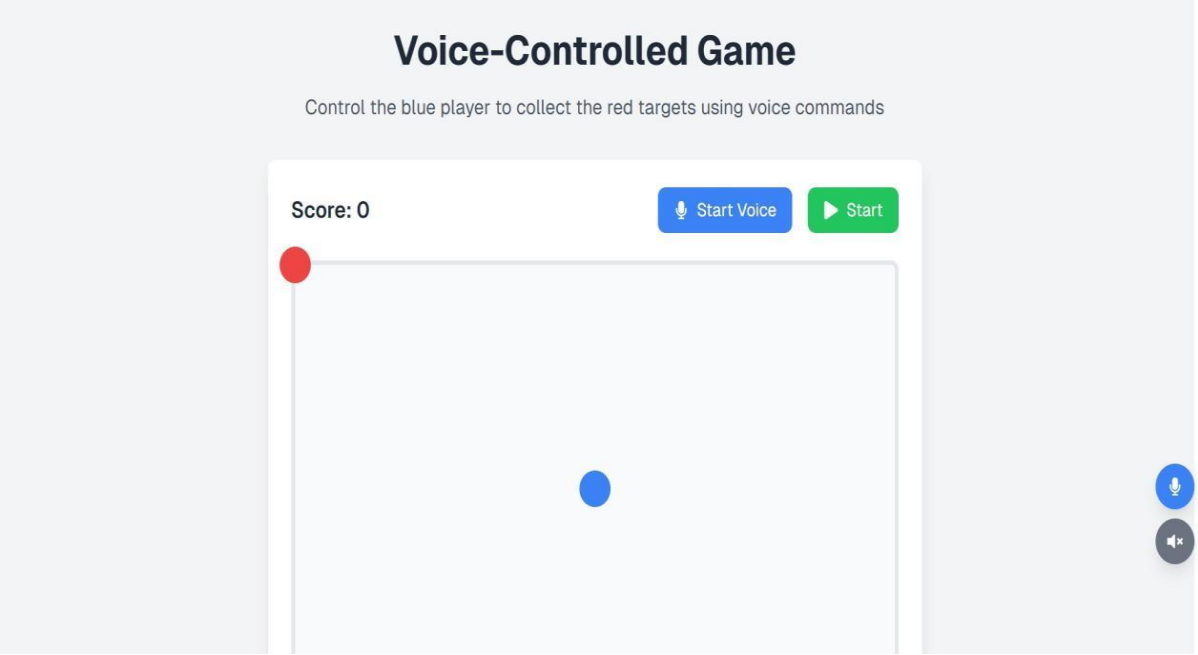
Quiz Game is an interactive, voice-enabled game designed to enhance learning through fun. Users can answer questions using voice commands, making it accessible and engaging. With instant feedback and scoring, it's perfect for quick knowledge checks and practice.





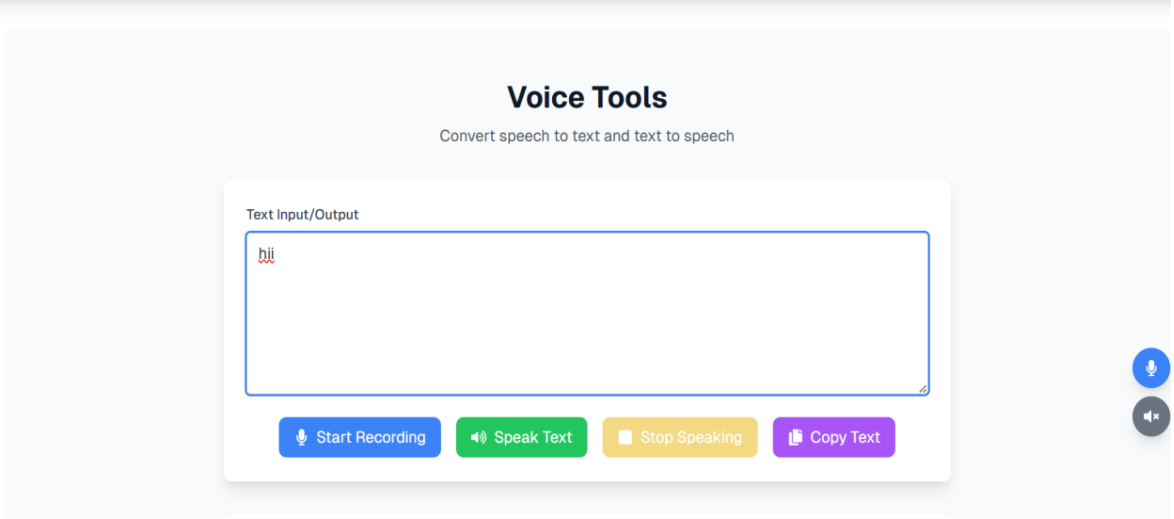
Voice Control Game

The **Voice Command Game** interface. It allows users to engage with a fun, accessible game controlled through voice input. The game provides an entertaining and interactive way for users with limited mobility to enjoy digital experiences without the need for physical controls.

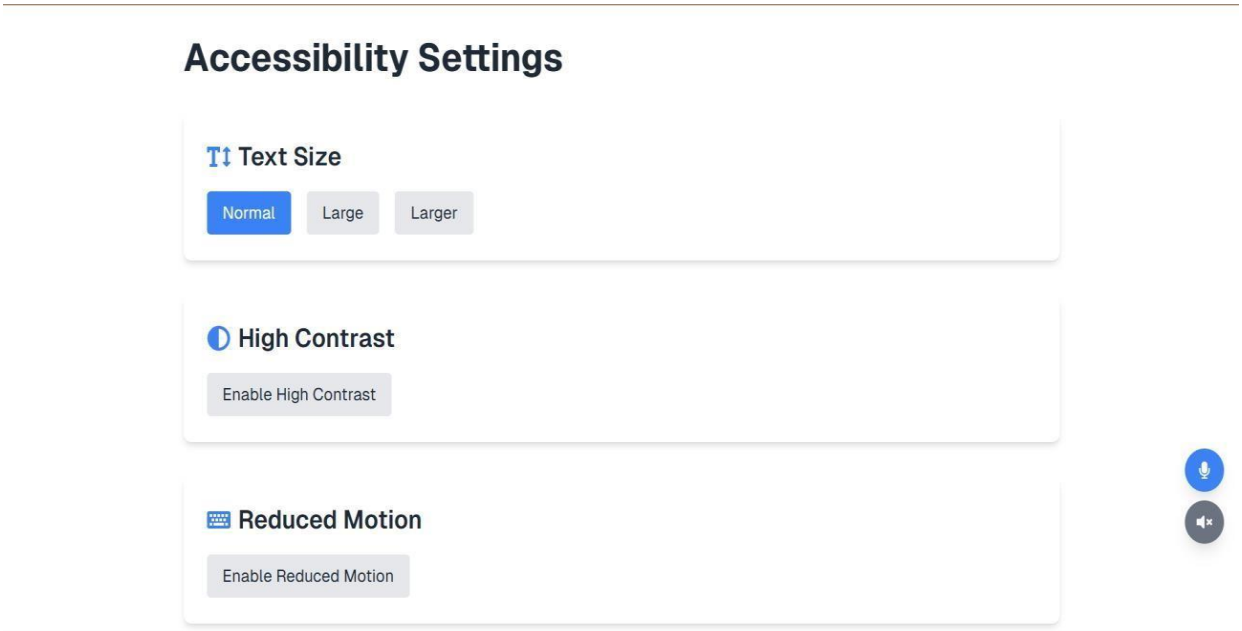


Voice Tools

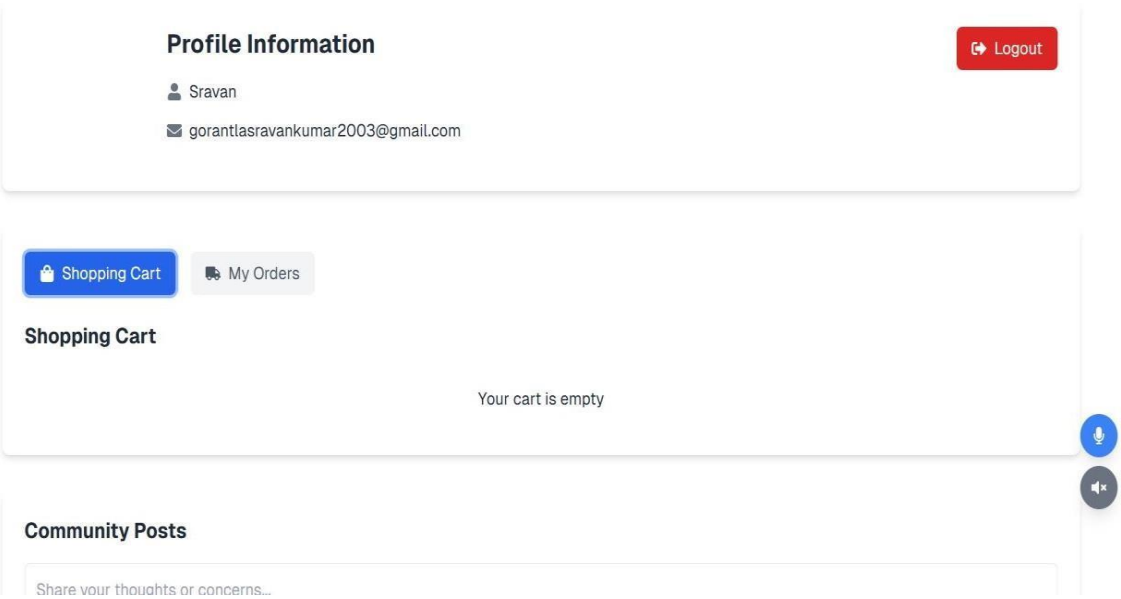
Here, the **Text-to-Speech and Voice-to-Text** functionality is highlighted. The tool converts user speech into text and vice versa, assisting visually impaired users or those with reading difficulties. The interface is user-friendly and supports natural interaction through voice commands.



Accessibility Settings



Profile Page



CHAPTER – 9
CONCLUSION

CONCLUSION

Navigating digital platforms remains a significant challenge for individuals with disabilities due to the limited availability of inclusive accessibility solutions. The Remote Accessibility Helper project is an innovative approach designed to overcome these barriers by providing a unified, intelligent, and user-centric platform. At its core, the system integrates a variety of accessibility tools—including a shopping cart and books section tailored with screen reader compatibility, an engaging voice command-based game, and real-time voice-to-text and text-to-speech conversion—to ensure smooth and intuitive interaction. Enhanced with screen reader support, dynamic text resizing, and contrast customization, it enables users to personalize their digital experience based on their specific needs. Built using Node.js, Express.js, and MongoDB, and secured with JWT-based authentication and disability ID verification, the system prioritizes both performance and data integrity. The platform's modular architecture supports scalable deployment across various devices and ensures accessibility even on low-end systems. By combining accessibility, interactivity, and security, the Remote Accessibility Helper doesn’t just assist—it empowers. It reimagines how individuals with visual, mobility, or cognitive challenges interact with technology, providing a gateway to digital independence and fostering a more inclusive online ecosystem.

CHAPTER – 10
BIBLIOGRAPHY

BIBLIOGRAPHY

W3 School – (https://www.w3schools.com/nodejs/)
Geeks for Geeks (https://www.geeksforgeeks.org/javascript_/)
NodeJS Official Documentation – (https://nodejs.org/docs/latest/api_/)
NextJS Official Documentation – (https://nextjs.com/en/starter/installing.html)
MongoDB – (https://www.mongodb.com/docs/manual/tutorial/)
React JS – (https://react.dev/learn)

CHAPTER – 11
REFERENCES

REFERENCES

[1]0020Mingzhe Li, Xiuying Chen, Weiheng Liao, Yang Song, Tao Zhang, Dongan Zhan, Rui Yan, “EZInterviewer: To Improve Job Interview Performance with Mock Interview Generator.” arXiv preprint arXiv:2301.00972, January 2023

[2] Fadel M. Megahed, Ying-Ju Chen, Joshua A. Ferris, Cameron Resatar, Kaitlyn Ross, Younghwa Lee, L. Allison Jones-Farmer, "ChatISA: A Prompt-Engineered Chatbot for Coding, Project Management, Interview and Exam Preparation Activities," *arXiv preprint arXiv:2407.15010*, June 2024.

[3] Taufiq Daryanto, Xiaohan Ding, Lance T. Wilhelm, Sophia Stil, Kirk McInnis Knutsen, Eugenia H. Rho, "Conversate: Supporting Reflective Learning in Interview Practice Through Interactive Simulation and Dialogic Feedback," *arXiv preprint arXiv:2410.05570*, October 2024.

[4] https://namastedev.com/learn/namaste-javascript

[5] https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Frameworks_libraries/React_getting_started

[6] https://libguides.usask.ca/understanding_genAI

[7] https://ai.google.dev/