SMART TEXT READER FOR THE VISUALLY IMPAIRED

PROJECT REPORT 2025-2026

Submitted By

24591643 SARGUNA

24507208 MANOJ

24507189 DIVYA

24507231 VIKASH

Under the guidance of

Ms.G.SUGANTHI B.E (CSE).,

LECT/COMPUTER

Diploma in Computer Engineering

of the Directorate of Technical Education, Government of Tamil Nadu

DEPARTMENT OF COMPUTER ENGINEERING

AKT MEMORIAL POLYTECHNIC COLLEGE

KALLAKURICHI-606202.

AKT MEMORIAL POLYTECHNIC COLLEGE

KALLAKURICHI-606202

Department of Computer Engineering

BONAFIDE CERTIFICATE

This is certified that this project work entitled **Smart Text Reader for the Visually Impaired** has been submitted **SARGUNA, MANOJ, DIVYA, VIKASH** in the partial fulfilment of the requirements for the award of Diploma in Computer Engineering during the academic year 2025-2026, who carried out the project work under our supervision.

Project Guide	**Head of the Department**
Ms.G.SUGANTHI B.E (CSE).,	**Mrs. C.SRIDEVI B.E.,(CSE)**
LECT/COMPUTER	**HOD-COMPUTER & IT**

This is to certify that_____ was examined for the project work viva-voce held on_____

INTERNAL EXAMINER EXTERNAL EXAMINER

Acknowledgement

We express our sincere and profound thanks to our chairman **Mr.A.K.T.Mahendiran B.Com.,** for creating this magnificent edifice of learning by providing all necessary facilities to complete diploma engineering course successfully.

We express our gratitude to our principal **Mr.P.K.Kabilar M.E.,MBA.,** for constant encouragement and facilities provided towards the successful completion of our project work.

At the same time we would like to express our heartfelt thanks to our head of the department **Mrs.C.Sridevi.,B.E.,(CSE).,** and also our project guide **Ms.G.Suganthi B.E (CSE).,** for guiding and needful advices and time to complete this project.

We would like to show our gratitude to our department staffs for all instructions and guidance given throughout.

Our sincere thanks and affection to our parents and friends who gave hand in all our steps.

CONTENTS

ABSTRACT

Smart Text Reader for the Visually Impaired is a useful and socially relevant final-year project developed to support visually impaired users in reading, understanding, and learning textual content through a digital platform. The main purpose of this system is to create an accessible environment where users can interact with text in a simple, structured, and user-friendly manner. In addition to text reading support, the project also focuses on braille learning, braille translation, assessment handling, progress monitoring, and feedback management. The complete system is designed using Core PHP and MySQL, making it cost-effective, easy to deploy, and suitable for academic demonstration as well as future real-world enhancement.

The proposed application helps visually impaired learners improve their literacy skills by combining text-based learning with braille-oriented features. It allows administrators to manage lessons, braille characters, translated content, tests, user records, performance reports, and feedback through a common professional interface. The system supports essential database functions such as create, read, update, and delete operations for all major modules, ensuring smooth content administration and data maintenance. This structured approach makes the project not only functional but also easy to manage, scalable, and presentation-ready for final-year academic evaluation.

A major strength of the system is its multi-functional learning support. Users can access learning materials, understand character mappings between normal text and braille, take assessments to evaluate their understanding, and track their progress over time. The translation module helps convert standard text into braille format, which improves familiarity with tactile reading concepts. Assessment and feedback modules further enhance the platform by helping administrators evaluate performance and collect user responses for continuous improvement. These combined features make the system more than a basic reader; it becomes a complete educational support tool for visually impaired learners.

The project is developed with a professional interface so that all operations can be managed in a centralized way. The admin panel can handle user management, content updates, learning resources, assessment questions, scores, progress reports, and feedback records efficiently. MySQL is used as the backend database to store and organize all application data securely, while Core PHP provides the logic required to connect modules and perform all system operations. This makes the application lightweight, practical, and suitable for implementation in institutions, training centers, or assistive learning environments.

1. INTRODUCTION

Smart Text Reader for the Visually Impaired is designed as an assistive web-based application that aims to improve accessibility, learning, and independence for visually impaired individuals. In today's digital era, most information is presented visually, creating a significant barrier for people who rely on alternative methods such as braille or audio. This project addresses that gap by providing a structured platform where users can read, learn, and interact with text using braille-based support and accessible system features. The system is developed using Core PHP and MySQL, ensuring simplicity, efficiency, and ease of deployment in academic and real-world environments.

The primary goal of this system is to assist visually impaired users in understanding textual information through a combination of braille learning and translation mechanisms. The application allows users to explore braille characters, understand their mapping with standard text, and practice reading through guided modules. By integrating a translation feature, the system converts regular text into braille format, enabling learners to gradually build familiarity and confidence. This makes the application not only a reading tool but also a complete educational platform for braille literacy.

In addition to learning and translation, the system includes assessment and evaluation features that help measure user progress. Users can take tests based on learned content, and their performance is recorded and analyzed within the system. Progress tracking modules provide insights into improvement over time, which is useful for both learners and administrators. This structured approach ensures that the system supports continuous learning rather than one-time interaction, making it more effective and engaging for users.

The application is built with a professional and centralized admin interface that manages all system functionalities. Administrators can control user data, learning materials, braille mappings, assessments, results, and feedback through a single dashboard. Each

module in the system supports full CRUD (Create, Read, Update, Delete) operations, which ensures flexibility and proper data management. The use of MySQL enables secure storage and organized handling of data, while Core PHP ensures smooth communication between different system components.

Furthermore, the system also includes a feedback management feature that allows users to share their experience, suggestions, and challenges. This helps in improving the system over time and making it more user-friendly. The overall design focuses on simplicity, accessibility, and usability so that even users with minimal technical knowledge can interact with the platform effectively. The interface is structured in a way that supports easy navigation and reduces complexity for both users and administrators.

2. LITERATURE SURVEY

. he literature related to assistive technology for visually impaired individuals shows a strong and growing interest in the development of digital systems that improve reading, learning, and communication accessibility. Traditional reading methods for visually impaired users have mainly depended on braille books, tactile charts, and human assistance. Although these methods remain important, they are often limited by availability, cost, portability, and update frequency. With the advancement of web technologies, educational software, and database-driven applications, researchers and developers have increasingly focused on building digital solutions that can support visually impaired learners in more interactive and scalable ways. This project, Smart Text Reader for the Visually Impaired, is inspired by these developments and aims to combine multiple assistive learning functions into one unified platform.

Many previous studies have emphasized the importance of braille literacy as a fundamental skill for visually impaired learners. Braille is not only a method of reading but also a tool for academic growth, communication, and independent living. However, braille learning often requires specialized materials and trained instructors, which may not always be accessible to every learner. Existing systems in the literature have attempted to solve this issue by introducing computerized braille learning modules, digital braille charts, and character-recognition-based training environments. These systems have shown that digital presentation of braille content can improve learning convenience and help users repeatedly practice symbols and patterns. The proposed project extends this idea by including braille learning as a structured module within a broader educational platform.

Another important area of research involves text-to-braille translation systems. Several studies have discussed automated translation techniques that convert regular text into braille representation using rule-based mappings and language-specific processing models. These tools are useful because they reduce the dependency on manually prepared braille content and enable users to access more learning resources. Some systems focus only on direct symbol conversion, while others attempt to improve

formatting, punctuation handling, and language accuracy. Although many of these systems achieve acceptable translation performance, they often lack integration with user learning environments, progress tracking, or assessment mechanisms. The current project addresses this gap by embedding translation functionality into a complete application that supports both learning and evaluation.

Research also shows the importance of assessment and performance analysis in educational systems designed for special-needs users. Evaluation modules help measure how effectively learners understand braille patterns, identify characters, and apply their knowledge in reading exercises. Existing literature on e-learning systems points out that assessments improve user engagement and provide measurable outcomes for both students and instructors. Some assistive systems include quizzes or exercises, but many are limited to standalone testing without long-term record management. The proposed Smart Text Reader improves this aspect by including assessments along with score storage, report generation, and progress tracking. This allows administrators to monitor user performance over time and identify areas where additional support may be needed.

Progress tracking is another concept widely discussed in learning management and assistive education research. Tracking user activities, completed lessons, test scores, and improvement trends helps create a personalized learning environment. Several studies on adaptive educational systems highlight that regular monitoring can motivate learners and make instruction more targeted. However, many simple assistive reading systems do not provide proper record maintenance or historical progress analysis. By integrating user progress modules with database storage, this project ensures that learning becomes measurable, structured, and transparent. This feature adds academic value to the system and makes it more suitable for institutional use.

Feedback collection has also been recognized in literature as an essential component of system improvement and user-centered design. In the case of visually impaired users, feedback becomes even more important because accessibility problems, navigation issues, and usability challenges may differ significantly from those of general users.

Existing research on accessible software design stresses that user opinion should be continuously gathered and analyzed to improve system effectiveness. Many earlier tools focus strongly on technical functions but ignore feedback handling. The proposed project includes a dedicated feedback management module so that administrators can review user suggestions, identify practical issues, and improve the platform over time.

In summary, the reviewed literature indicates that assistive systems for visually impaired individuals have evolved from isolated tools into more integrated digital learning environments. Braille learning modules, text-to-braille translation, performance assessment, progress monitoring, and feedback analysis have each been studied as valuable components, but they are often implemented separately. The Smart Text Reader for the Visually Impaired brings these essential elements together in one web-based application using Core PHP and MySQL. By combining accessibility support with educational management and CRUD-based administration, the project builds upon existing research while offering a more practical, unified, and academically relevant solution for final-year implementation.

3. PROPOSED SYSTEM

The proposed system, Smart Text Reader for the Visually Impaired, is a web-based assistive learning and management platform developed to support visually impaired users in reading, learning braille, practicing translation, attending assessments, tracking performance, and submitting feedback. The system is designed using Core PHP for application logic and MySQL for database management. Its main objective is to create an accessible and organized digital environment where visually impaired learners can improve their literacy skills and interact with textual content in a more meaningful way. Unlike basic reading tools, this proposed system combines educational support and administrative control within a single professional interface.

The system is structured to provide both user-side learning features and admin-side management features. On the user side, learners can access braille lessons, understand character representations, use text-to-braille translation tools, attend assessments, and view their own learning progress. On the admin side, the system offers a centralized dashboard to manage users, braille content, learning materials, assessments, results, progress records, and feedback. All important modules are connected through a common interface, and each major table supports Create, Read, Update, and Delete (CRUD) operations. This ensures easy data maintenance, proper content control, and smooth system administration.

A major component of the proposed system is the braille learning module, which introduces visually impaired users to the structure and meaning of braille characters. The system stores braille symbols, alphabets, words, and learning content in the database, allowing administrators to update and organize educational materials whenever required. Learners can use this module to understand the relationship between standard text and braille forms. By presenting the content in a structured way, the system helps users gradually improve their familiarity with braille reading concepts. This makes the platform suitable for beginners as well as learners who need continuous practice.

Another important feature is the text-to-braille translation module, which converts regular text into braille representation using stored mappings and predefined conversion logic. This feature allows users to enter text and receive corresponding braille output, which supports both learning and practice. The translation process is designed to be simple, efficient, and understandable, so that users can quickly recognize how each letter or word is represented in braille format. Since the translation records can also be stored and managed, the system creates a reusable learning database that strengthens educational value. This module acts as a bridge between conventional text and accessible reading methods.

The proposed system also includes an assessment and evaluation module to measure user understanding and learning performance. After completing braille lessons or translation practice, users can attend quizzes or assessments prepared by the administrator. The system stores questions, answers, scores, and performance details in the database. Once the test is completed, results are automatically processed and displayed through the platform. This helps learners identify their strengths and weaknesses while also enabling administrators to monitor academic performance. The assessment module transforms the project from a simple content viewer into an interactive learning solution.

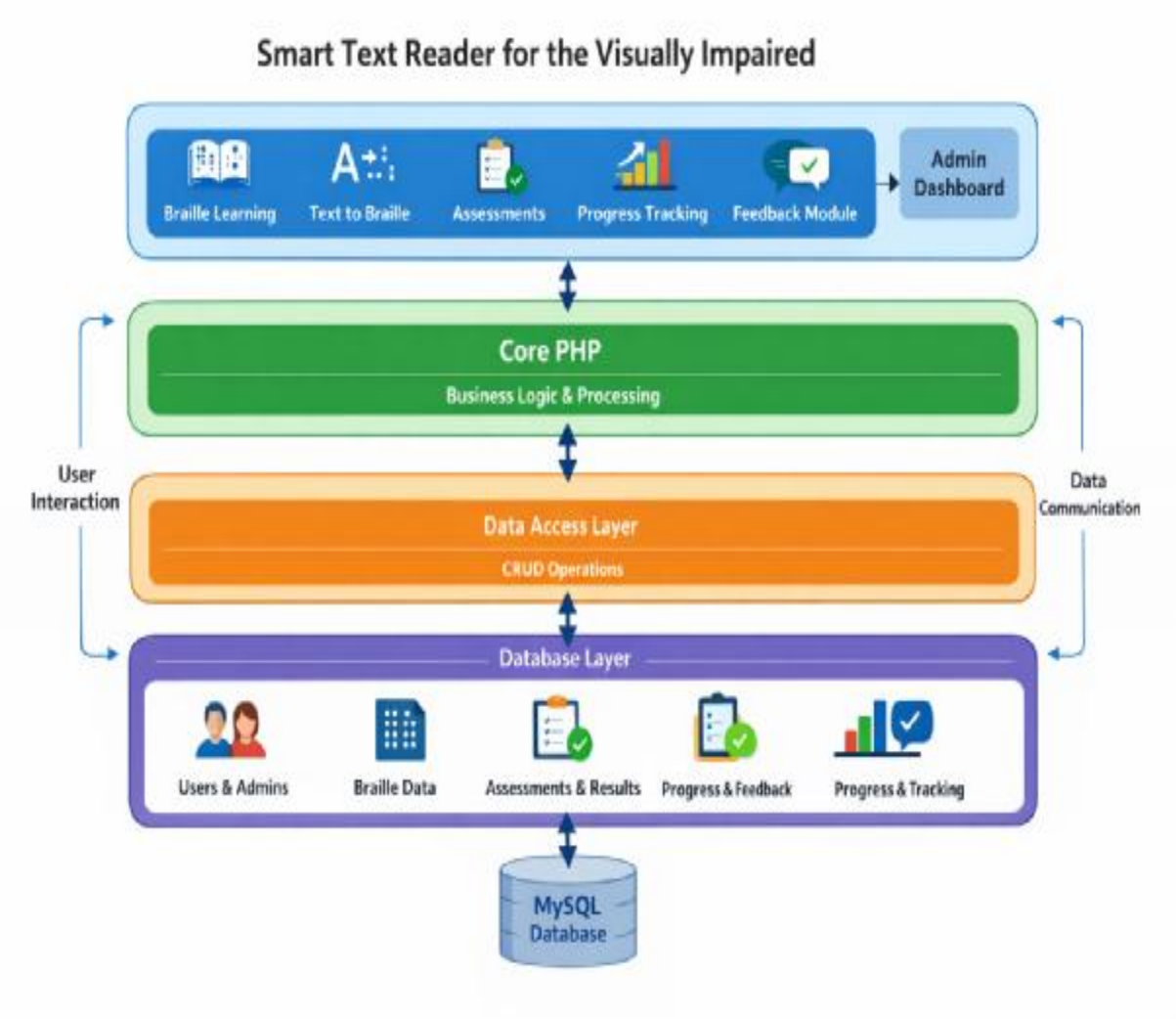
To support continuous improvement, a progress tracking module is integrated into the system. This module records user activities such as completed lessons, assessment scores, learning milestones, and historical improvement. Progress reports can be viewed through the dashboard, enabling both learners and administrators to understand development over time. Such tracking is useful for identifying which topics require additional attention and for motivating users by showing measurable achievements. This feature gives the system an educational management dimension that is often missing in basic accessibility tools.

The system further includes a feedback management module, which allows users to share opinions, suggestions, and issues regarding the platform. This feature is essential

because accessibility applications must be continuously refined based on real user experience. Administrators can read, manage, and respond to feedback through the same common interface used for other modules. In addition, the system provides user management, report generation, and content maintenance functionalities that make the entire platform complete and professionally structured. Since all major tables are CRUD-enabled, the administrator can easily perform insertion, modification, deletion, and retrieval of records without difficulty.

Overall, the proposed Smart Text Reader for the Visually Impaired is an integrated and practical system that combines accessibility, education, and management in one platform. It improves upon traditional methods by offering braille learning, translation, assessments, progress monitoring, and feedback handling in a database-driven web application. The use of Core PHP and MySQL makes the system lightweight, affordable, and easy to implement in schools, training centers, and academic project environments. The proposed model not only supports visually impaired users in learning and reading but also demonstrates how modern web technology can contribute to inclusive education and digital empowerment.

3.1 ARCHITECTURE



The architecture of the Smart Text Reader for the Visually Impaired is designed as a layered web-based system that integrates user interaction, application processing, database management, and administrative control into a single structured platform. The system is developed using Core PHP as the server-side scripting language and MySQL as the backend database. Its architecture is planned in such a way that it supports accessibility-oriented learning features such as braille learning, text-to-braille translation, assessments, progress tracking, and feedback handling, while also providing a professional interface for managing all records through common CRUD operations. This architecture ensures that the project remains simple, modular, scalable, and suitable for final-year academic implementation.

At the top level, the system contains the presentation layer, which represents the front-end interface used by both learners and administrators. This layer includes login pages, dashboards, forms, tables, reports, learning screens, translation inputs, quiz interfaces, and feedback forms. The interface is designed to be simple, organized, and professional so that users can navigate the system without confusion. For visually impaired learning support, the user-side interface focuses on structured content access, readable layouts, and easy interaction with learning materials. The administrator-side interface provides options to add, edit, update, delete, and monitor all records such as braille lessons, user details, assessment questions, scores, and feedback. This layer acts as the communication bridge between the end user and the internal logic of the system.

The second layer is the application or business logic layer, which is responsible for processing user requests and controlling the system workflow. This is the core functional layer of the project and is implemented using Core PHP. Whenever a user logs in, requests a braille lesson, submits text for translation, takes an assessment, or sends feedback, this layer receives the request, validates the input, processes the required logic, and sends the result back to the interface. For example, if a learner enters normal text into the translation module, the business logic layer retrieves the corresponding braille mapping rules, converts the text into braille format, and displays the translated result. Similarly, when users attend an assessment, this layer checks the answers, calculates the score, stores the result, and updates the progress record. This layer is important because it controls how each module behaves and ensures that all components work together properly.

The next part of the architecture is the data access layer, which manages communication between the PHP-based logic and the MySQL database. This layer handles all database queries related to inserting, reading, updating, and deleting records. Since the project supports CRUD operations for all major tables, the data access layer plays a central role in maintaining system data. It stores and retrieves information related to users, braille symbols, lessons, translated text, questions, answers, scores, progress details, and feedback. By separating data operations from the presentation layer, the architecture

becomes more organized and easier to maintain. This layer also improves consistency, because all modules use the same structured approach for database interaction.

At the bottom of the architecture lies the database layer, where MySQL is used to store all application information in a structured relational format. Different tables are maintained for users, admin details, braille content, text translations, assessments, question banks, results, progress tracking, and feedback. Each table is linked logically based on the relationships required by the application. For example, a user table may be connected with assessment results and progress history, while the braille mapping table is connected with translation processes. Because the system is database-driven, all learning records and management activities are stored permanently, making report generation and future analysis easier. The database layer ensures data consistency, organized storage, and reliable retrieval of information whenever needed.

The architecture also includes module-based functional flow, where each system module works as an independent unit but remains connected with the overall platform. The login and authentication module verifies user identity before granting access. The braille learning module provides educational content and stored symbol patterns. The translation module converts normal text into braille using stored mappings. The assessment module delivers quiz questions and evaluates responses. The progress tracking module stores learning performance over time. The feedback module records user suggestions and issues. The admin management module connects all these parts by giving the administrator complete control over data and operations. This modular architecture improves clarity, simplifies development, and allows future expansion of the project.

From a working perspective, the architecture follows a request-response model. First, the user interacts with the front-end interface and submits a request, such as viewing a lesson, translating text, or taking a test. The request is then passed to the PHP logic layer, where validation and processing take place. If the request needs data, the logic layer communicates with the MySQL database through the data access layer. After

fetching or updating records, the processed result is returned to the front-end and displayed to the user. This repeated workflow ensures smooth interaction across all modules. For administrative operations, the same cycle is followed, but with advanced management privileges such as adding new lessons, editing quiz questions, or deleting feedback records.

Security and control are also considered within the architecture. Authentication mechanisms ensure that only authorized administrators can access management modules, while users can only access permitted learning functions. Input validation is handled in the PHP layer to reduce errors and improve data integrity. Database operations are structured carefully so that records remain consistent and easy to manage. Since all major tables are CRUD-enabled, the architecture supports efficient maintenance without requiring major technical complexity. This makes the project highly practical for educational institutions and accessible technology demonstrations.

3.2 MODULES DESCRIPTION

1. User Management Module

The User Management Module is one of the most important components of the Smart Text Reader for the Visually Impaired system. This module is responsible for maintaining the complete details of users who access the application. It stores information such as user name, email, contact details, login credentials, role type, and account status. Since the system may contain both administrators and learners, this module helps in controlling access based on authorization level. The administrator can add new users, edit existing user details, update account information, and delete inactive records whenever necessary. This module supports full CRUD operations and ensures that all user-related data is stored systematically in the MySQL database.

This module also manages secure login and authentication features. When a user enters login credentials, the system verifies the details and grants access only if the account is valid. By separating users and administrators, the module helps maintain security and proper workflow. Learners can access only the educational and personal progress features, while administrators can access management functions such as content creation, assessment configuration, and feedback monitoring. This role-based management improves the reliability of the system and prevents unauthorized data modification.

Another major advantage of this module is that it supports easy profile management. Users can update their basic details, and administrators can monitor account activity through a centralized interface. The module also becomes the foundation for linking other parts of the system, because user records are connected to assessments, progress reports, translations, and feedback history. In short, the User Management Module acts as the entry point and identity control center of the entire application.

2. Admin Dashboard Module

The Admin Dashboard Module provides a centralized and professional interface through which all core operations of the system can be managed. It acts as the control panel for administrators and displays quick access to major modules such as user

management, braille learning content, translation records, assessments, progress reports, and feedback details. This module is designed to make administration simple, fast, and organized. Instead of managing each section separately in an unstructured way, the dashboard brings all functions into one place, improving system efficiency and usability.

Through this module, the administrator can monitor the overall performance of the application. Summary details such as total users, number of lessons, completed assessments, feedback count, and progress records can be viewed in a single screen. This gives a clear understanding of how the system is functioning and helps in quick decision-making. The dashboard may also display alerts or notifications related to newly submitted feedback, pending content updates, or recently registered users. Such visual summaries make the system more professional and easy to present in a final-year project demonstration.

The Admin Dashboard Module also serves as the navigation layer for all CRUD-based operations. The administrator can move from one module to another without confusion and manage every table from a common interface. This module improves data control, reduces complexity, and ensures that the entire platform remains user-friendly for system administrators. In short, it acts as the operational heart of the project.

3. Braille Learning Module

The Braille Learning Module is the educational core of the Smart Text Reader for the Visually Impaired system. Its main purpose is to help visually impaired learners understand and practice braille symbols in a structured digital format. This module stores information related to braille alphabets, numbers, special symbols, and word patterns in the database. The content can be organized lesson by lesson so that learners can begin with basic characters and gradually move toward advanced braille reading concepts. This structured presentation supports effective learning and improves accessibility.

The administrator can use this module to add new lessons, update character mappings, edit descriptions, and delete outdated or incorrect records. Since the module supports CRUD operations, educational content can be continuously improved without affecting

the rest of the application. Learners can access available lessons and review braille patterns repeatedly, which is an essential part of skill development. The digital format also reduces dependency on physical learning materials and makes braille education easier to maintain in one online platform.

This module plays an important role in building the learner's confidence. By studying braille through organized lessons, users become more familiar with the connection between ordinary text and braille representation. The module supports continuous practice, repeated learning, and concept clarity. It serves as the foundation for the translation and assessment modules, because users first need to understand braille symbols before using advanced features. Therefore, the Braille Learning Module is central to the educational value of the project.

4. Text-to-Braille Translation Module

The Text-to-Braille Translation Module is one of the key functional modules in the system. This module allows users to enter regular text and receive the corresponding braille output based on stored mappings in the database. The main objective is to create an easy bridge between standard text and braille learning so that users can understand how letters, words, and symbols are represented in braille form. This conversion process supports both learning and practical usage, making the module highly valuable in the overall system.

The module works by taking user input, processing it through application logic written in Core PHP, and matching each character with the braille pattern stored in the database. Once the conversion is complete, the translated output is displayed on the screen. Administrators can manage braille mappings through CRUD operations, which means they can insert new mappings, update existing patterns, or remove invalid entries. This flexibility helps improve system accuracy and ensures better translation quality.
Another useful aspect of this module is that translated records can be stored for reference, review, or reporting. Learners can use the translation tool repeatedly to practice and compare text forms. This increases engagement and helps users become more comfortable with braille recognition. The module is simple in operation but

powerful in educational impact. It transforms the project from a static learning platform into an interactive assistive application with real conversion capability.

5. Lesson Content Management Module

The Lesson Content Management Module is designed to organize and maintain all educational materials used in the system. While the Braille Learning Module focuses on the learning interface, this module handles the backend management of lessons, topics, descriptions, examples, and practice content. It allows administrators to create structured learning units that can be easily accessed by users. For example, different categories such as alphabets, words, punctuation, and number representations can be created and maintained through this module.

The administrator can add new lesson content, edit descriptions, update examples, and delete outdated materials whenever necessary. Since the project is database-driven, every lesson is stored systematically in MySQL and retrieved through the Core PHP interface. This ensures proper organization and fast content delivery. It also makes the system scalable, because new lessons can be added at any time without affecting existing modules. The professional interface allows content management to be done smoothly and efficiently.

This module improves the quality of learning by ensuring that educational materials remain clear, updated, and well arranged. It also supports user progress because learners can move through lessons in a logical order. The existence of a dedicated content management module makes the project stronger from an academic point of view, as it shows proper separation between content creation and content delivery. In short, it ensures that the learning experience remains rich, consistent, and manageable.

6. Assessment Module

The Assessment Module is designed to evaluate the learning level of users after they complete braille lessons or translation practice. This module includes quizzes, test questions, answer options, score calculation, and result generation. It helps measure whether learners are correctly understanding braille symbols, recognizing translations,

and applying what they have studied. This makes the system more interactive and educational rather than just content-based. Assessments give both learners and administrators a clear idea of academic progress.

Administrators can create assessments through the common management interface by adding questions, correct answers, difficulty level, and category details. These questions are stored in the database and can be modified using CRUD operations. Users can attempt the assessments through a dedicated interface, and once they submit their answers, the system automatically processes the result. Scores are then calculated and stored in the database for future analysis. This automation reduces manual work and ensures consistent evaluation.

The Assessment Module also increases learner engagement by introducing a challenge-based approach to education. Users can test their knowledge, identify mistakes, and improve weak areas. For administrators, the module provides a measurable way to understand how effective the teaching content is. This makes the platform more dynamic and suitable for institutional use. In summary, the assessment module is an essential part of turning the Smart Text Reader into a complete digital learning solution.

7. Results and Progress Tracking Module

The Results and Progress Tracking Module is responsible for storing and analyzing the performance of users over time. This module takes information from completed assessments and learning activities and converts it into meaningful progress records. It helps learners understand how much they have improved and helps administrators evaluate the effectiveness of the system. Progress tracking is very important in educational platforms because learning becomes more successful when improvement is visible and measurable.

This module stores information such as completed lessons, assessment scores, dates of participation, progress percentage, and performance history. Through the admin panel, these records can be viewed, filtered, updated, or deleted if needed. Users may also be allowed to view their own progress reports, giving them motivation to continue

learning. Because the project uses MySQL for data storage, historical records remain محفوظ and can be used for future analysis and reporting.

The biggest strength of this module is that it provides continuity in the learning process. Instead of treating each activity as separate, the system connects them and shows how the learner is developing over time. This is especially useful in visually impaired education, where repeated practice and gradual mastery are essential. The progress tracking feature makes the project more intelligent, organized, and educationally meaningful.

8. Feedback Management Module

The Feedback Management Module allows users to provide their opinions, suggestions, and issues regarding the system. This module is essential because assistive applications must be continuously improved based on real user experience. Visually impaired learners may face practical challenges while using the platform, and feedback helps administrators understand those issues clearly. By providing a formal mechanism for users to communicate their experiences, the system becomes more user-centered and adaptable.

Users can submit feedback through a form interface, and the submitted records are stored in the database. Administrators can view all feedback messages, update their status, respond internally, or remove irrelevant entries through CRUD operations. This makes feedback management systematic rather than informal. It also helps during project presentations, because it shows that the application is designed not only for service delivery but also for continuous improvement.

This module contributes to the long-term quality of the platform. Suggestions from users can lead to interface changes, improved lesson organization, better translation logic, or enhanced accessibility features. The feedback module therefore acts as a communication bridge between learners and administrators. It strengthens the practical value of the project and demonstrates a commitment to inclusive and responsive system design.

9. Reports Module

The Reports Module is designed to generate meaningful summaries from system data for analysis and administration. This module can produce reports related to users, braille lessons, assessment results, progress history, and feedback records. Reports help administrators monitor system performance in a structured way and are also useful during final-year project demonstrations. Instead of checking raw database records one by one, reports present organized information that supports decision-making and evaluation.

Through this module, administrators can view user-wise performance, lesson completion details, assessment score summaries, and feedback statistics. Reports may be displayed in table format, and filters can be used to search or sort data. Since the system is built with Core PHP and MySQL, report data is fetched dynamically from the database and shown in a clear manner. Administrators may also use these reports to identify weak learners, frequently attempted lessons, or commonly reported issues.

The Reports Module improves the professional quality of the project by adding an analytical dimension. It transforms stored data into useful information. This makes the system more than just a transactional platform; it becomes a management and review tool as well. In conclusion, the reports module adds administrative value, academic usefulness, and clarity to system operations.

10. Database and CRUD Management Module

The Database and CRUD Management Module is the backbone of the entire Smart Text Reader for the Visually Impaired system. Every major feature in the project depends on proper data storage, retrieval, modification, and deletion. This module ensures that all core tables such as users, admins, braille mappings, lessons, assessments, results, progress records, and feedback are connected to a common professional interface. It supports Create, Read, Update, and Delete operations across the entire system.

Whenever the administrator inserts a new lesson, edits a braille mapping, updates user information, deletes feedback, or reviews an assessment result, this module handles the required database communication. Core PHP processes the request, and MySQL

stores or retrieves the necessary data. Because all important tables are CRUD-enabled, the project becomes flexible, maintainable, and easy to demonstrate in a final-year setting. This common approach also reduces coding complexity and ensures consistency across modules

4. IMPLEMENTATION

The implementation of the Smart Text Reader for the Visually Impaired is carried out using a structured and modular approach with Core PHP as the backend scripting language and MySQL as the relational database. The system is deployed on a web server environment such as Apache (XAMPP/WAMP or VPS hosting), where all modules are integrated into a single application. The implementation focuses on simplicity, efficiency, and maintainability, ensuring that each component of the system works seamlessly while supporting accessibility-oriented features like braille learning, translation, assessment, progress tracking, and feedback management.

The development begins with database design, where multiple tables are created to store system data in an organized manner. Important tables include users, admin, braille mappings, lesson content, translations, assessments, questions, results, progress tracking, and feedback. Each table is structured with appropriate fields, primary keys, and relationships to ensure data consistency and easy retrieval. MySQL is used for its reliability, speed, and compatibility with PHP. Proper normalization is followed to reduce redundancy and maintain data integrity across all modules.

The backend implementation is handled using Core PHP, where each module is developed as a separate file or set of files to maintain modularity. PHP scripts are used to handle form submissions, process user inputs, connect with the database, and execute CRUD operations. For example, when a user registers or logs in, PHP validates the credentials and interacts with the user table. Similarly, when an administrator adds a braille lesson or updates a question, PHP scripts process the request and update the database accordingly. Reusable functions are created for database connections, query execution, and input validation to reduce code duplication and improve efficiency.

The frontend interface is designed using HTML, CSS, and basic JavaScript to create a clean and user-friendly layout. The system includes separate interfaces for users and administrators. The user interface provides access to learning modules, translation tools, assessments, progress reports, and feedback forms. The admin interface includes a

professional dashboard with menu-based navigation for managing all system components. The design focuses on clarity, proper alignment, and easy navigation so that users can interact with the system without confusion. Although the system is simple, it maintains a professional appearance suitable for academic presentation.

The authentication and security implementation ensures that only authorized users can access specific modules. Login functionality is implemented using PHP sessions, which store user identity after successful authentication. Passwords can be stored securely using hashing techniques to prevent unauthorized access. Role-based access control is applied so that administrators can manage data while regular users can only access learning features. Input validation and basic security checks are included to prevent invalid data entry and improve system reliability.

The braille learning and translation features are implemented using database-driven logic. Braille symbols and mappings are stored in a dedicated table, and PHP scripts retrieve these mappings when required. In the translation module, user input text is processed character by character, and each letter is matched with its corresponding braille pattern. The result is then displayed on the screen. This implementation ensures that translation is dynamic and can be easily updated by modifying database records rather than changing code.

The assessment module implementation includes creating question banks, storing correct answers, and designing quiz interfaces. When users attempt an assessment, their responses are captured and processed using PHP logic. The system compares user answers with correct answers, calculates the score, and stores the result in the database. This result is then linked with the user profile and used for progress tracking. The implementation ensures automatic evaluation, which reduces manual work and provides instant feedback to users.

The progress tracking feature is implemented by storing user activity data such as completed lessons, test scores, and performance history. PHP scripts retrieve this data

and display it in a structured format, such as tables or simple reports. This allows users to monitor their improvement over time. Administrators can also view overall system performance and identify users who may need additional support. This module adds analytical value to the system and enhances the learning experience.

The feedback module implementation allows users to submit comments, suggestions, or issues through a form interface. The submitted data is stored in the feedback table and displayed in the admin dashboard for review. Administrators can manage feedback records using CRUD operations, which helps in maintaining a responsive and user-centered system. This module ensures that the system can evolve based on real user needs and experiences.

4.1 SOFTWARE REQUIEMENTS

1. Operating System

- Windows 10 / 11 (for development)
- Linux (Ubuntu/CentOS) recommended for production server

2. Web Server

- Apache Server (with mod_rewrite enabled)

  OR

- Nginx Server

3. Backend Technology

- PHP >= 8.2
- Laravel Framework (Latest Version)

4. Database

- MySQL Server (5.7 or higher)

  OR

- MariaDB

5. Frontend Technology

- HTML5
- CSS3
- JavaScript
- Bootstrap (for responsive UI design)
- Blade Template Engine (Laravel)

6. Required PHP Extensions

- PDO PHP Extension
- OpenSSL PHP Extension
- Mbstring PHP Extension
- Exif PHP Extension
- Fileinfo Extension
- XML PHP Extension
- Ctype PHP Extension
- JSON PHP Extension
- Tokenizer PHP Extension

- cURL PHP Extension

7. Additional Tools & Software

- Composer (Dependency Manager for PHP)
- Git (Version Control System)
- Node.js & NPM (for frontend asset compilation, optional)

8. Server Configuration

- Enable **mod_rewrite** (Apache)
- Enable **HTTPS (SSL Certificate)** for secure communication
- Configure .env file for database and app settings
- Set proper file permissions for storage and cache

9. Browser Compatibility

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Safari

10. Hosting Environment

- VPS Server / Cloud Hosting (AWS, DigitalOcean, etc.)
- Minimum recommended configuration:
    - 2+ CPU Cores
    - 4 GB RAM
    - 50 GB Storage

4.2 SYSTEM REQUIREMENTS

1. Hardware Requirements

Development System Requirements

These are the minimum hardware requirements for developing and testing the project:

- **Processor:** Intel Core i3 / i5 or higher
- **RAM:** 8 GB minimum
- **Hard Disk:** 256 GB SSD or higher
- **Monitor:** 14-inch or above
- **Keyboard and Mouse:** Standard input devices
- **Internet Connection:** Stable broadband connection

These specifications are sufficient for coding, database handling, local server testing, and UI development.

Server Requirements (VPS Hosting)

For live deployment, the application requires a **Virtual Private Server (VPS)** with the following configuration:

Component	Specification
Server Type	Virtual Private Server (VPS)
CPU	8 Core Processor
RAM	32 GB
Storage	300 GB NVMe SSD
Bandwidth	High-speed / Unlimited preferred
Operating System	Ubuntu / CentOS / AlmaLinux
Web Server	Apache or Nginx
Database Server	MariaDB
Control Panel	WHM / cPanel
Backup Support	Daily / Weekly Backup Recommended
SSL Certificate	Required for secure access

4.3 SAMPLE CODING

Dashboard

```
<?php
require_once 'includes/auth.php';
require_once 'config/db.php';
$pageTitle = 'Dashboard';
$pageSubtitle = 'Overview of the braille training project modules';
include 'includes/header.php';

$counts = [
  'roles' => $pdo->query('SELECT COUNT(*) FROM roles')->fetchColumn(),
  'users' => $pdo->query('SELECT COUNT(*) FROM users')->fetchColumn(),
  'students' => $pdo->query('SELECT COUNT(*) FROM students')->fetchColumn(),
  'characters' => $pdo->query('SELECT COUNT(*) FROM braille_characters')->fetchColumn(),
  'translations' => $pdo->query('SELECT COUNT(*) FROM translations')->fetchColumn(),
  'practice' => $pdo->query('SELECT COUNT(*) FROM practice_sessions')->fetchColumn(),
  'quizzes' => $pdo->query('SELECT COUNT(*) FROM quizzes')->fetchColumn(),
  'results' => $pdo->query('SELECT COUNT(*) FROM quiz_results')->fetchColumn(),
  'progress' => $pdo->query('SELECT COUNT(*) FROM progress_logs')->fetchColumn(),
  'feedback' => $pdo->query('SELECT COUNT(*) FROM feedback')->fetchColumn(),
];
$recentTranslations = $pdo->query('SELECT t.id, t.input_text, t.translated_output, u.full_name, t.created_at FROM translations t JOIN users u ON u.id = t.translated_by ORDER BY t.id DESC LIMIT 5')->fetchAll();
?>
```

```
<section class="grid">
  <?php foreach ($counts as $label => $value): ?>
    <div class="card metric-card">
      <div class="metric-label"><?= e(ucwords(str_replace('_', ' ', $label))) ?></div>
      <div class="metric-value"><?= e((string)$value) ?></div>
      <div class="metric-note">Current records in <?= e($label) ?></div>
    </div>
  <?php endforeach; ?>
</section>
<section class="section-grid two-col">
  <div class="card">
    <h3>Project Summary</h3>
    <p>This final-year project demonstrates braille learning, translation, assessments, progress tracking, and feedback management using Core PHP and MySQL. All major tables support create, read, update, and delete operations through a common professional interface.</p>
    <div class="badge-list">
      <span class="badge">10 Tables</span>
      <span class="badge">10 CRUD Modules</span>
      <span class="badge">Reusable Layout</span>
      <span class="badge">REST API</span>
    </div>
  </div>
  <div class="card">
    <h3>Quick Actions</h3>
    <div class="quick-links">
      <a href="modules/students.php" class="quick-link">-ЬjЊ Manage Students</a>
      <a href="modules/braille_master.php" class="quick-link">Paҷ Manage Characters</a>
```

```
        <a href="modules/translations.php" class="quick-link">-ЂЋЂ Open Translator</a>
        <a href="modules/reports.php" class="quick-link">-ЂЊі View Reports</a>
        <a href="modules/print_report.php" class="quick-link">-Ђќе Print Report</a>
      </div>
    </div>
</section>
<section class="card">
    <h3>Recent Translation History</h3>
    <div class="table-wrap">
      <table>
        <thead><tr><th>ID</th><th>Input</th><th>Output</th><th>Translated By</th><th>Date</th></tr></thead>
        <tbody>
        <?php foreach ($recentTranslations as $row): ?>
          <tr>
            <td><?= e((string)$row['id']) ?></td>
            <td><?= e($row['input_text']) ?></td>
            <td><?= e($row['translated_output']) ?></td>
            <td><?= e($row['full_name']) ?></td>
            <td><?= e($row['created_at']) ?></td>
          </tr>
        <?php endforeach; ?>
        </tbody>
      </table>
    </div>
</section>

<div class="card convention-card">
    <div class="toolbar">
      <div>
```

```
        <h3>Braille Alphabet Convention</h3>
        <p class="small">The project now includes the same alphabet convention reference shown in the provided image.</p>
      </div>
      <a class="btn btn-light" href="modules/braille_master.php">Open Braille Master</a>
    </div>
    <img src="assets/images/braille-alphabet-convention.png" alt="Braille alphabet convention" class="convention-image dashboard-convention-image">
</div>

<?php include 'includes/footer.php'; ?>
```

Login

```
<?php
require_once __DIR__ . '/includes/app.php';
require_once __DIR__ . '/config/db.php';

if (isset($_SESSION['user'])) {
    redirect_to('dashboard.php');
}

if (is_post()) {
    $email = trim($_POST['email'] ?? '');
    $password = $_POST['password'] ?? '';

    $stmt = $pdo->prepare('SELECT u.*, r.role_name FROM users u JOIN roles r ON
r.id = u.role_id WHERE email = ? LIMIT 1');
    $stmt->execute([$email]);
    $user = $stmt->fetch();

    if ($user && password_verify($password, $user['password'])) {
        $_SESSION['user'] = $user;
        set_flash('success', 'Welcome back, ' . $user['full_name'] . '.');
        redirect_to('dashboard.php');
    }

    set_flash('error', 'Invalid email or password.');
    redirect_to('login.php');
}
$flash = get_flash();
?>
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login - Braille PHP Project</title>
    <link rel="stylesheet" href="assets/css/style.css">
</head>
<body class="login-body">
    <div class="login-layout">
        <div class="login-panel login-hero">
            <span class="hero-chip">Final Year Main Project</span>
            <h1>Braille Learning and Translation Management System</h1>
            <p>A professional Core PHP application with complete CRUD for braille training, translation history, assessments, feedback, and reports.</p>
            <ul class="feature-list">
                <li>Professional sidebar dashboard</li>
                <li>10+ CRUD modules</li>
                <li>Ready SQL files with 10 tables</li>
            </ul>
        </div>
        <div class="login-panel">
            <h2>Sign in</h2>
            <p class="muted">Use the demo admin account to enter the project.</p>
            <?php if ($flash): ?>
                <div class="alert alert-<?= e($flash['type']) ?>"><?= e($flash['message']) ?></div>
            <?php endif; ?>
            <form method="post" class="stack-form">
                <div>
                    <label>Email</label>
                    <input type="email" name="email" value="admin@brailleproject.com" required>
```

```
            </div>
            <div>
                <label>Password</label>
                <input type="password" name="password" value="admin123" required>
            </div>
            <button type="submit">Login</button>
        </form>
        <div class="demo-box">
            <strong>Demo Login</strong>
            <span>admin@brailleproject.com / admin123</span>
        </div>
    </div>
  </div>
</body>
</html>
```

Master File

```
<?php
require_once '../includes/auth.php';
require_once '../config/db.php';
require_once '../includes/crud.php';

$pageTitle = 'Braille Character Master';
$pageSubtitle = 'Manage braille characters with full CRUD and standard alphabet convention reference';

if (isset($_GET['delete'])) {
    delete_by_id($pdo, 'braille_characters', (int)$_GET['delete']);
```

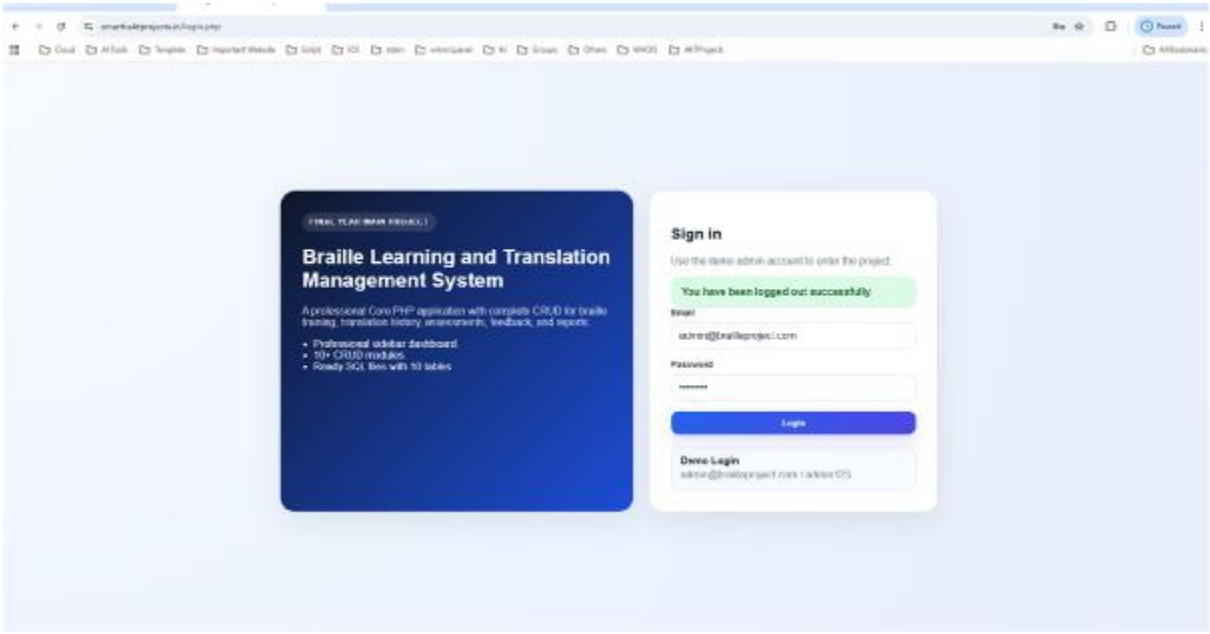
```
    set_flash('success', 'Record deleted successfully.');
    redirect_to('braille_master.php');
}

$edit = null;
if (isset($_GET['edit'])) {
    $stmt = $pdo->prepare('SELECT * FROM braille_characters WHERE id = ?');
    $stmt->execute([(int)$_GET['edit']]);
    $edit = $stmt->fetch();
}

if (is_post()) {
    $id = (int)($_POST['id'] ?? 0);
    $alphabet = trim($_POST['alphabet'] ?? '');
    $code = trim($_POST['braille_code'] ?? '');
    $description = trim($_POST['description'] ?? '');
    $values = [$alphabet, $code, $description];

    if ($id > 0) {
        $stmt = $pdo->prepare('UPDATE braille_characters SET alphabet = ?, braille_code = ?, description = ? WHERE id = ?');
        $values[] = $id;
        $stmt->execute($values);
        set_flash('success', 'Record updated successfully.');
    } else {
        $stmt = $pdo->prepare('INSERT INTO braille_characters (alphabet, braille_code, description) VALUES (?, ?, ?)');
        $stmt->execute($values);
        set_flash('success', 'Record added successfully.');
    }
    redirect_to('braille_master.php');
```

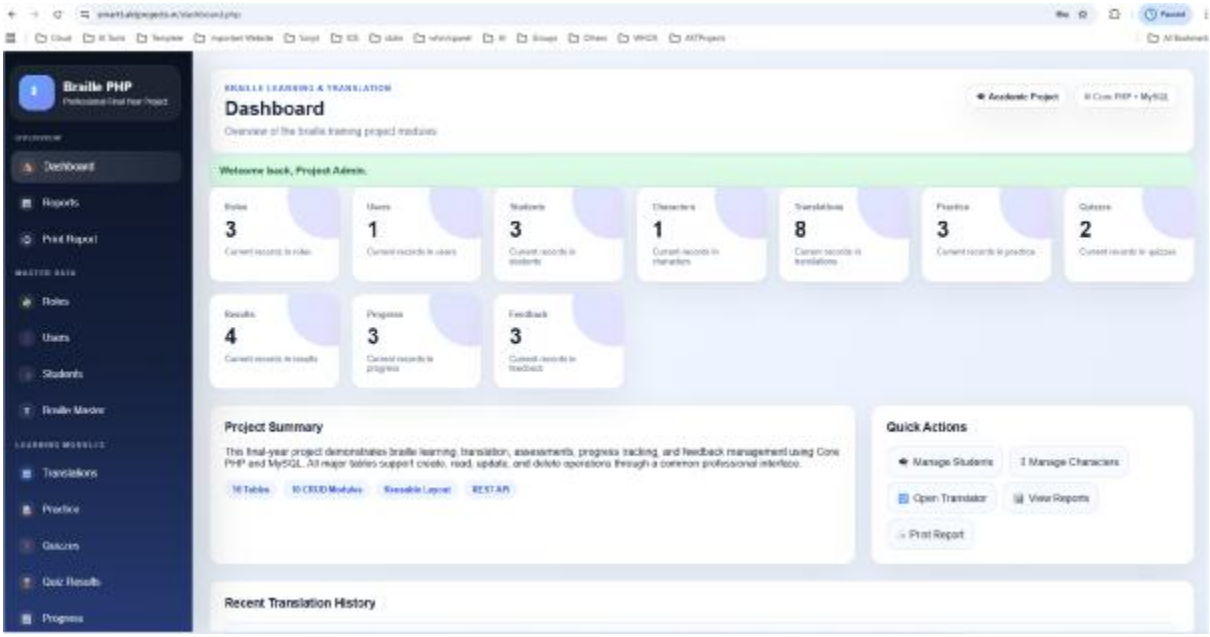
```
}

$rows = $pdo->query('SELECT * FROM braille_characters ORDER BY alphabet ASC')->fetchAll();
include '../includes/header.php';
?>
<div class="section-grid two-col">
  <div class="card">
    <h3><?= $edit ? 'Edit Record' : 'Add New Record' ?></h3>
    <form method="post" class="form-grid">
      <input type="hidden" name="id" value="<?= e((string)($edit['id'] ?? '')) ?>">

      <div>
        <label>Alphabet</label>
        <input type="text" name="alphabet" maxlength="5" value="<?= e($edit['alphabet'] ?? '') ?>" required>
      </div>
      <div>
        <label>Braille Code</label>
        <input type="text" name="braille_code" placeholder="Example: 100000" value="<?= e($edit['braille_code'] ?? '') ?>" required>
      </div>
      <div style="grid-column:1/-1">
        <label>Description</label>
        <textarea name="description" rows="3"><?= e($edit['description'] ?? '') ?></textarea>
      </div>
      <div class="actions">
        <button type="submit"><?= $edit ? 'Update Record' : 'Add Record' ?></button>
        <a class="btn btn-light" href="braille_master.php">Reset</a>
```

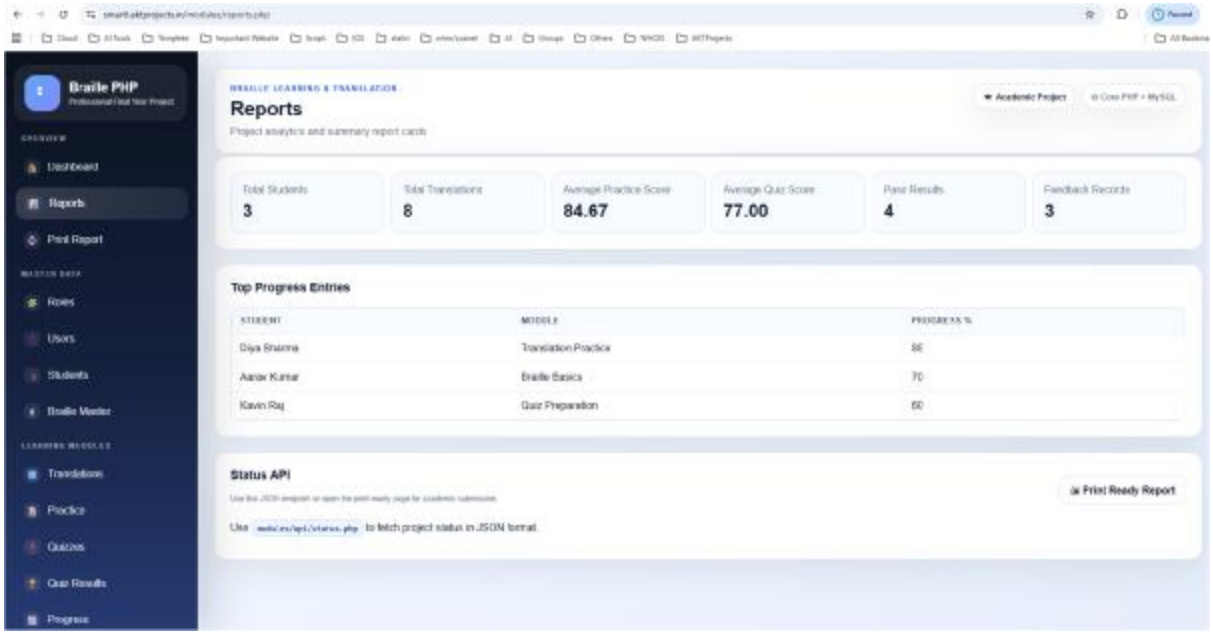
```
        </div>
      </form>
    </div>

    <div class="card convention-card">
      <h3>Braille Alphabet Convention</h3>
      <p class="muted">Reference chart added as per the provided image. Use this standard convention while entering or validating braille patterns.</p>
      <img src="../assets/images/braille-alphabet-convention.png" alt="Braille alphabet convention" class="convention-image">
      <div class="badge-list">
        <span class="badge">a-z reference included</span>
        <span class="badge">Standard 6-dot pattern</span>
        <span class="badge">Useful for data entry</span>
      </div>
    </div>
  </div>

  <div class="card">
    <div class="toolbar">
      <div>
        <h3>Display Records</h3>
        <p class="small">Current master contains the full alphabet convention for translation and learning modules.</p>
      </div>
      <div class="actions">
        <span class="status-pill muted-pill">Total Records: <?= count($rows) ?></span>
      </div>
    </div>
    <div class="table-wrap">
```

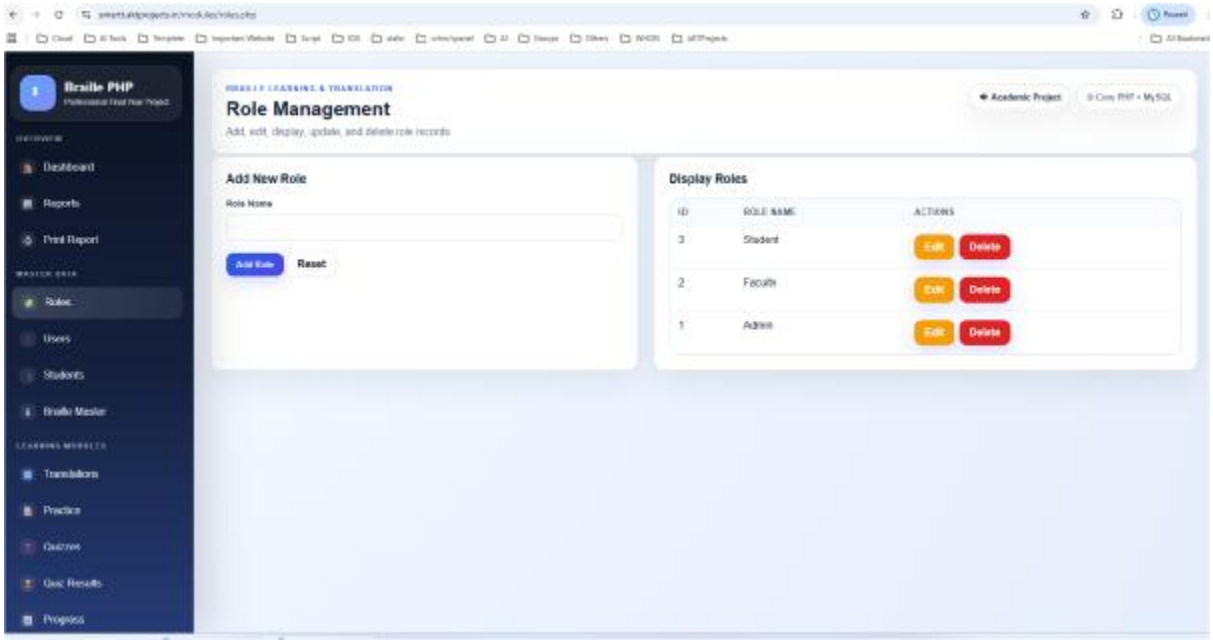
```
    <table>
      <thead>
      <tr>
        <th>ID</th>
        <th>Alphabet</th>
        <th>Braille Code</th>
        <th>Description</th>
        <th>Actions</th>
      </tr>
      </thead>
      <tbody>
      <?php foreach ($rows as $row): ?>
        <tr>
          <td><?= $row['id'] ?></td>
          <td><?= e($row['alphabet']) ?></td>
          <td><code><?= e($row['braille_code']) ?></code></td>
          <td><?= e($row['description']) ?></td>
          <td class="action-links">
            <a class="btn btn-warning" href="braille_master.php?edit=<?= $row['id'] ?>">Edit</a>
            <a class="btn btn-danger" data-confirm="Delete this record?" href="braille_master.php?delete=<?= $row['id'] ?>">Delete</a>
          </td>
        </tr>
      <?php endforeach; ?>
      </tbody>
    </table>
  </div>
</div>
<?php include '../includes/footer.php'; ?>
```

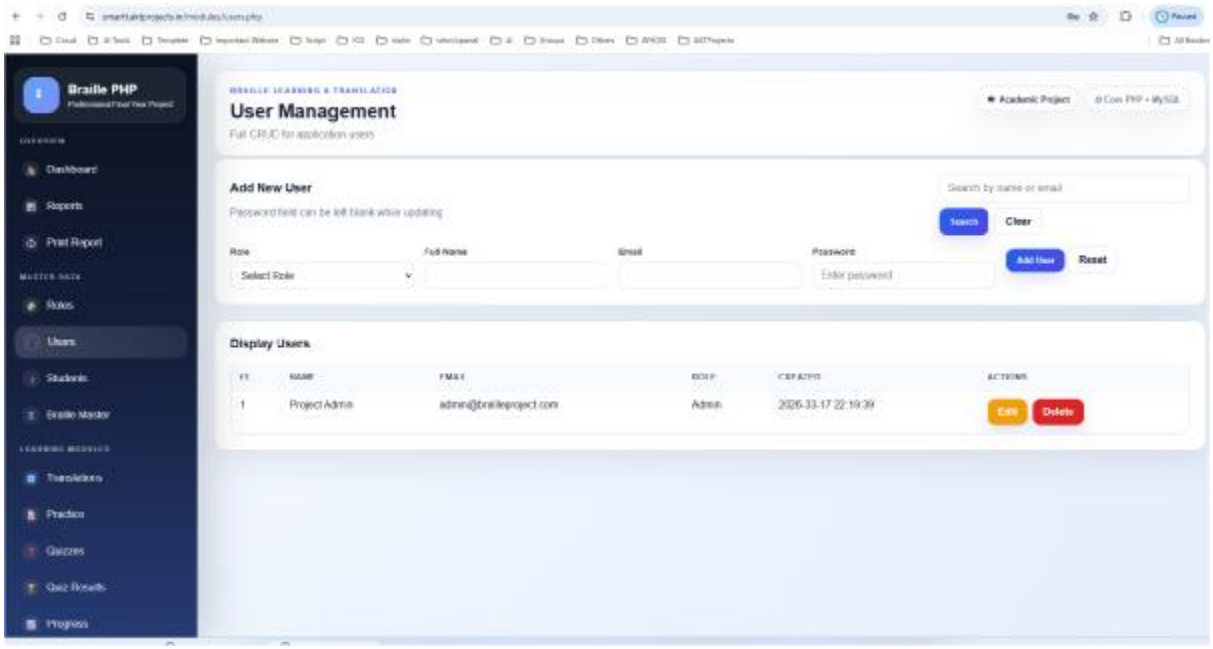
4.4. SCREEN SHOT

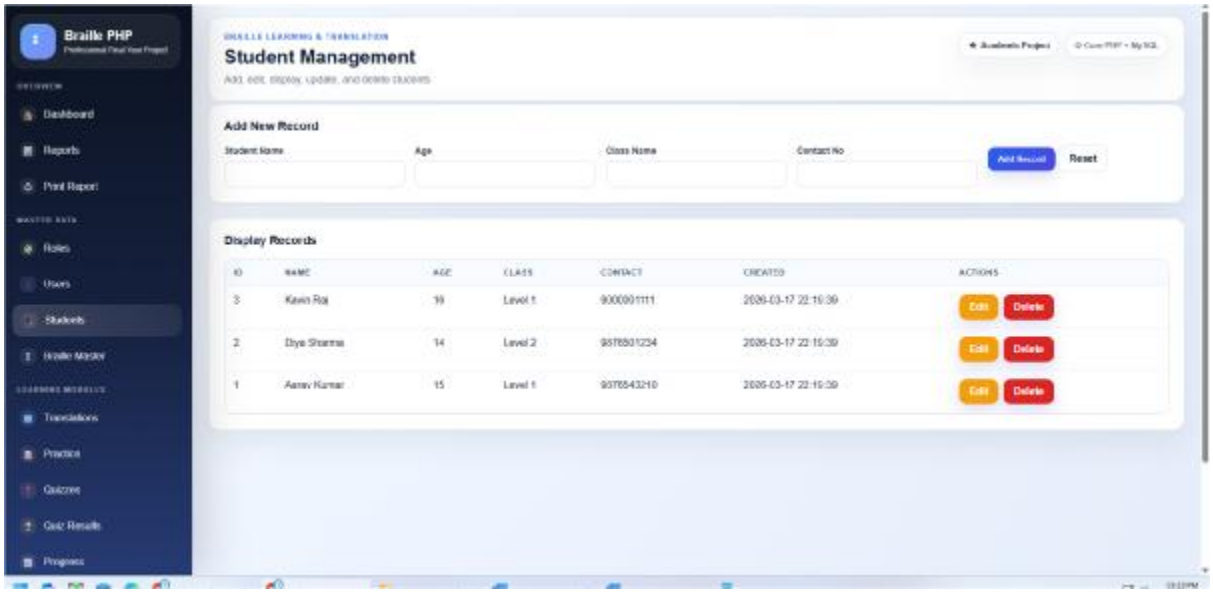


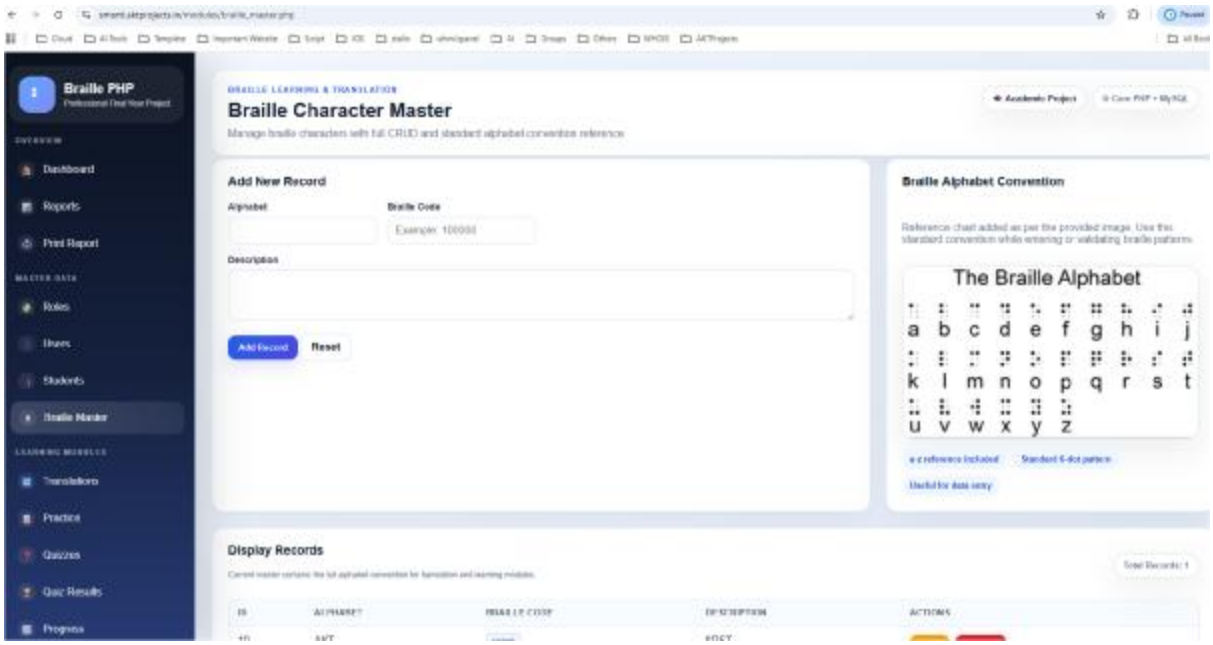


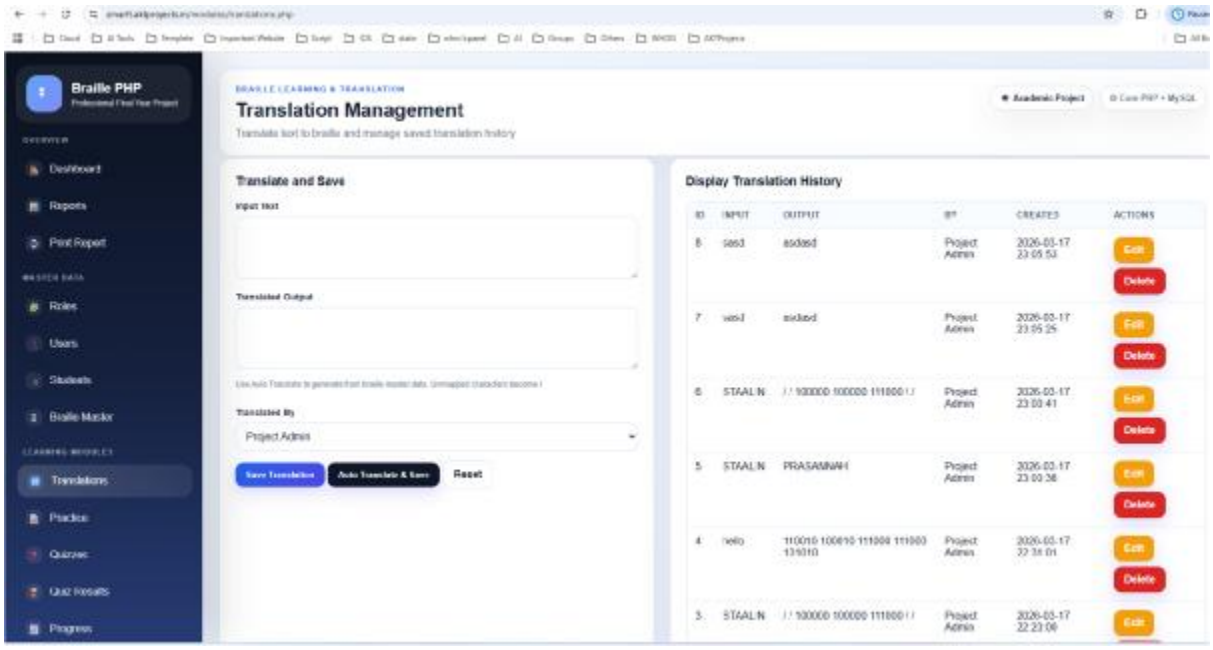

Braille PHP
Dashboard
Reports
Print Report
Roles
Users
Students
Braille Master
Translations
Practice
Quizzes
Quiz Results
Progress
Reports
Total Students
3
Total Translations
8
Average Practice Score
84.67
Average Quiz Score
77.00
Pass Results
4
Feedback Records
3
Top Progress Entries
Diya Sharma
Translation Practice
Aarav Kumar
Braille Basics
70
Kavin Raj
Quiz Preparation
60
Status API
Print Ready Report


Braille PHP
Dashboard
Reports
Print Report
Roles
Users
Students
Braille Master
Translations
Practice
Quizzes
Quiz Results
Progress
Role Management
Add New Role
Role Name
Add Role
Reset
Display Roles
ID
ROLE NAME
ACTIONS
3
Student
Edit
Delete
2
Faculty
Edit
Delete
1
Admin
Edit
Delete


Braille PHP
User Management
Full CRUD for application users
Add New User
Password field can be left blank while updating
Search by name or email
Search
Clear
Role
Full Name
Email
Password
Select Role
Enter password
Add User
Reset
Display Users
ID
NAME
EMAIL
ROLE
CREATED
ACTIONS
1
Project Admin
admin@brailleproject.com
Admin
2026-03-17 22:19:39
Edit
Delete
Dashboard
Reports
Print Report
Roles
Users
Students
Braille Master
Translations
Practice
Quizzes
Quiz Results
Progress


Braille PHP
Student Management
Add New Record
Student Name
Age
Class Name
Contact No
Add Record
Reset
Display Records
ID
NAME
AGE
CLASS
CONTACT
CREATED
ACTIONS
3
Kevin Raj
19
Level 1
9000001111
2026-03-17 22:19:39
Edit
Delete
2
Diya Sharma
14
Level 2
9876501234
2026-03-17 22:19:39
Edit
Delete
1
Aarav Kumar
15
Level 1
9876543210
2026-03-17 22:19:39
Edit
Delete
Dashboard
Reports
Print Report
Roles
Users
Students
Braille Master
Translations
Practice
Quizzes
Quiz Results
Progress


Braille PHP
Braille Character Master
Add New Record
Braille Alphabet Convention
The Braille Alphabet
Display Records


Braille PHP
Translation Management
Translate and Save
Display Translation History

5. CONCLUSION

The Smart Text Reader for the Visually Impaired is a meaningful and practical final-year project that successfully combines accessibility, education, and web-based technology into a single integrated system. The project addresses a real-world challenge faced by visually impaired individuals by providing a digital platform that supports braille learning, text-to-braille translation, assessments, progress tracking, and feedback management. By using Core PHP and MySQL, the system is designed to be simple, efficient, and easy to deploy, making it suitable for academic demonstration as well as future real-world applications.

One of the major achievements of this project is its ability to bring multiple functionalities together in a structured and user-friendly interface. Instead of focusing only on text reading, the system extends its capabilities to include learning support, performance evaluation, and continuous improvement through feedback. The inclusion of CRUD operations across all modules ensures that data management is flexible and well-organized. This makes the system highly maintainable and scalable, allowing administrators to easily manage users, content, assessments, and reports from a centralized dashboard.

The project also highlights the importance of inclusive technology in modern society. By creating a platform that supports visually impaired learners, it demonstrates how web applications can be designed to serve diverse user needs. Features like braille learning and translation not only improve accessibility but also promote independence and self-learning among users. The assessment and progress tracking modules further enhance the system by providing measurable outcomes, which are essential for educational growth and evaluation.

From a technical perspective, the project showcases the effective use of Core PHP for backend processing and MySQL for structured data storage. The modular architecture ensures that each component functions independently while remaining connected

within the overall system. The implementation of authentication, role-based access, and database-driven operations reflects good development practices. The project also maintains a professional interface, making it suitable for presentation and demonstration in academic environments.

REFERENCES

1. World Health Organization (WHO), World Report on Vision, Geneva, 2019.
2. American Foundation for the Blind (AFB), “Braille Literacy Resources,” Available: https://www.afb.org
3. National Federation of the Blind (NFB), “Importance of Braille Literacy,” Available: https://www.nfb.org
4. R. W. Mann, “Braille Translation and Communication Systems,” Journal of Visual Impairment & Blindness, vol. 102, no. 5, pp. 267–278, 2018.
5. S. K. Jain and P. K. Singh, “Assistive Technologies for Visually Impaired: A Review,” International Journal of Computer Applications, vol. 150, no. 10, pp. 15–20, 2016.
6. M. A. Hersh and M. A. Johnson, Assistive Technology for Visually Impaired and Blind People, Springer, 2008.
7. PHP Documentation, “PHP: Hypertext Preprocessor,” Available: https://www.php.net
8. Oracle Corporation, “MySQL Reference Manual,” Available: https://dev.mysql.com/doc
9. W3C, “Web Content Accessibility Guidelines (WCAG) 2.1,” Available: https://www.w3.org/TR/WCAG21/
10. J. Lazar, D. Goldstein, and A. Taylor, Ensuring Digital Accessibility through Process and Policy, Morgan Kaufmann, 2015.
11. R. Pressman, Software Engineering: A Practitioner’s Approach, McGraw-Hill, 8th Edition, 2014.
12. I. Sommerville, Software Engineering, Pearson Education, 10th Edition, 2016.
13. A. Dix, J. Finlay, G. Abowd, and R. Beale, Human-Computer Interaction, Pearson, 3rd Edition, 2004.
14. IEEE Xplore Digital Library, “Research Papers on Assistive Technologies,” Available: https://ieeexplore.ieee.org
15. Google Scholar, “Studies on Braille Learning and Assistive Systems,” Available: https://scholar.google.com