

NATURAL LANGUAGE PROCESSING

NLP-Based Student Readability Assessment and Personalized Feedback System

B. Kiran, S. Chaya, N M. Divya, K N. Manohar

Department of Computer Science and Engineering, Mysuru Royal Institute of Technology, Mysuru, Karnataka, India

Published October 2026

Abstract

Student writing assessment is important in academic learning, but manual evaluation of written responses can require considerable time and may provide limited information about specific writing difficulties. This paper presents an NLP-Based Student Readability Assessment and Personalized Feedback System for automated analysis of student English writing. The system allows users to enter text directly or upload PDF, DOCX, and TXT documents. Submitted content is analyzed for readability, grammar, spelling, vocabulary, sentence structure, and text statistics. Seven readability measures are used to estimate reading difficulty, while vocabulary richness, repeated words, difficult words, sentence length, and sentence type provide additional writing characteristics. The system combines these results to generate an overall writing-quality score and a feature-weighted proficiency estimate. Personalized suggestions explain detected weaknesses and recommend areas for improvement. Teacher-managed assignments with reference answers and student submissions are also supported. Analysis records are stored per user in a lightweight database.

Keywords: student writing assessment; readability analysis; natural language processing; personalized feedback; grammar analysis; vocabulary analysis; educational technology

1 Introduction

English writing is an important part of academic learning because students regularly prepare assignments, examination answers, essays, reports, and other written responses. The quality of such writing depends on several characteristics, including readability, grammar, spelling, vocabulary usage, sentence construction, and clarity. Students may find it difficult to identify these weaknesses on their own, while manual evaluation of many submissions can require significant time and may produce only a final score without explaining the factors that affected the result.

Automated text analysis can support this process by examining measurable characteristics of written content and presenting the results in a structured form. Readability formulas can estimate the difficulty of a text, while lexical and sentence-level measures can provide additional information about vocabulary use and sentence construction. Rule-based checking can identify detectable grammar and spelling issues, allowing the student to review specific problems instead of relying only on an overall mark.

The proposed NLP-Based Student Readability Assessment and Personalized Feedback System is a web-based application developed to combine these analysis functions in a single workflow. Students can enter writing directly or upload supported PDF, DOCX, and TXT documents. The system extracts readable text and processes it through readability, grammar, spelling, vocabulary, and sentence-

analysis components. The resulting measures are used to calculate an overall writing-quality score and to generate improvement suggestions.

The system also includes a teacher-oriented assignment workflow. A teacher can create an assignment with instructions, a minimum word requirement, and a reference answer. Students can submit answers and receive a comparison based on textual similarity and covered concepts, while teachers can view submissions associated with their assignments. User authentication and user-specific storage are used to keep analysis history and assignment information organized.

An additional feature provides a transparent feature-weighted regression baseline for estimating writing proficiency. The baseline combines measurable writing features such as readability, grammar, vocabulary, sentence structure, clarity, vocabulary richness, error rate, average sentence length, and complex-sentence share. The implementation treats this model as an academic demonstration rather than a validated research-grade predictor, so no unsupported accuracy claim is made.

1.1 Objectives

- To develop a web-based system for automated assessment of student English writing.
- To calculate readability using multiple established readability measures.
- To analyze grammar, spelling, vocabulary, sentence structure, and text statistics.
- To generate an overall writing-quality score and explainable assessment factors.
- To provide personalized suggestions based on the detected writing characteristics.
- To support teacher assignments, reference-answer comparison, and student submissions.

2 Literature survey

Research in automated writing assessment has investigated readability prediction, automated essay scoring, grammar correction, and feedback generation as separate or combined tasks. Flesch introduced a widely used readability measure based on sentence length and word characteristics [1]. Kincaid et al. developed the Flesch-Kincaid readability formulas for assessing the reading level of written material [2]. Gunning proposed the Fog Index for estimating the educational level required to understand a text [3], while McLaughlin introduced the SMOG approach for estimating readability from polysyllabic words [4]. Coleman and Liau proposed a readability formula based on characters and sentences rather than syllable counts [5]. Dale and Chall developed a word-list-based readability formula that also considers sentence length [6].

Automated essay scoring research has increasingly used statistical and machine-learning features to estimate writing quality. Modern studies also investigate deep learning and large language models for scoring and feedback generation. These approaches motivate the use of measurable writing characteristics while also highlighting the importance of evaluation datasets and validation procedures. The present project focuses on a lightweight, interpretable implementation suitable for an academic web application and combines readability with grammar, spelling, vocabulary, sentence analysis, assignment comparison, and feedback. General NLP concepts and information-retrieval methods provide the foundation for text processing and similarity calculation [7], [8].

3 Methodology

The implemented system follows a sequential web-based workflow. A user first registers or logs in as a student or teacher. Authentication is handled through password hashing and JSON Web Tokens, after which the available functions are controlled according to the user's role. For writing analysis, the student enters text directly or uploads a PDF, DOCX, or TXT file. The server extracts readable text in memory before passing it to the analysis pipeline. The current implementation accepts files up to 10 MB and does not perform OCR on scanned or image-only PDFs.

The extracted text is converted into words, sentences, and paragraphs and basic statistics are calculated. Readability is then evaluated using Flesch Reading Ease, Flesch-Kincaid Grade Level, Gunning Fog Index, SMOG Index, Coleman-Liau Index, Automated Readability Index, and Dale-Chall Score. Grammar checking uses rule-based patterns for detectable issues such as repeated words, lowercase pronouns, extra spacing, missing spaces after commas, and sentence-capitalization problems. Spelling is checked separately using a bundled offline dictionary, common-error corrections, and edit-distance-based suggestions.

Vocabulary analysis calculates vocabulary richness, unique words, average word length, simple-word count, difficult-word count, and repeated-word information. Sentence analysis identifies simple, compound, and complex sentences and flags long and very long sentences. These results are combined into a writing-quality score. The system also sends selected features to a transparent feature-weighted regression baseline to estimate a proficiency level.

Finally, the system generates strengths, improvement suggestions, and an explainable summary of the factors contributing to the assessment. Results are stored in the local JSON-backed database. Assignment submissions use term-frequency vectors and cosine similarity for reference-answer comparison, together with key concept coverage and writing-quality measures.

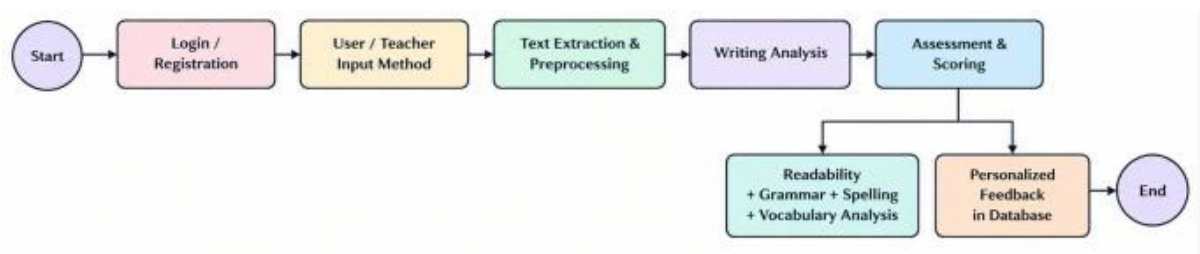


Figure 1. Flowchart of the proposed student writing assessment and feedback system.

4 Modeling and analysis

The system is organized into presentation, authentication, analysis, assessment, assignment, and storage functions. The client interface provides login, writing input, document upload, analysis results, history, and assignment pages. The server provides authentication, document extraction, spelling detection, analysis storage, assignment management, reference-answer comparison, and proficiency prediction endpoints. The database stores users, analyses, assignments, and submissions.

The analysis pipeline is designed so that each measurable writing characteristic is calculated independently before the final assessment is produced. Readability scores describe text difficulty, grammar and spelling checks identify detectable errors, vocabulary measures describe lexical variety, and sentence analysis describes sentence length and structural type. The quality-score component combines readability, grammar, vocabulary, and sentence-structure measures. The proficiency

baseline then uses a separate set of normalized features and produces a score, level, and six-point equivalent.

Student Readability Assessment Home Writing Analysis Reference Comparison Progress About student: Student Logout

Writing analysis

Edit your text below, then run a full analysis covering readability, statistics, sentence structure, vocabulary, grammar and overall writing quality.

Technology has become an important part of modern education. It has changed the way students learn and how educational content is delivered. In the past, students mainly depended on textbooks and classroom teaching. Today, computers, smartphones, online learning platforms, digital libraries, and educational applications provide students with many additional learning resources.

One of the main benefits of technology in education is easy access to information. Students can search for information, watch educational videos, attend online classes, and use digital learning materials from almost anywhere. Technology also allows students to learn at their own pace. Difficult topics can be studied again using recorded lectures, interactive lessons, and online

Upload PDF / DOCX / TXT No file selected

Analyze Clear

Words **269**

Characters **1987**

Sentences **21**

Paragraphs **6**

Reading time **1 min**

Figure 2. Writing analysis and assessment interface.

Table 1. System modules and their functions.

Module	Function
Authentication	Registration, login, password hashing, JWT sessions, and role checks.
Input & Upload	Direct text entry and PDF, DOCX, or TXT document extraction.
Readability	Seven readability formulas and reading-difficulty indicators.
Grammar & Spelling	Rule-based grammar checks and offline dictionary-based spelling suggestions.
Vocabulary	Richness, unique words, average word length, difficult and repeated words.
Sentence Analysis	Sentence type, length, and long/very-long sentence identification.
Scoring & Feedback	Writing-quality score, proficiency baseline, explanations, and suggestions.
Assignments	Teacher-created tasks, reference answers, student submissions, and comparison.
Storage	User-specific analyses, assignments, and submissions in the local database.

The application layer communicates with backend services through defined API operations. Authentication services validate users and issue sessions. Document extraction prepares supported files for

analysis, while the writing-analysis functions calculate the measurable text characteristics. The assessment layer combines selected measures and generates feedback. Assignment services provide teacher-created reference tasks and compare student answers. The database stores the records required by these functions.

5 Proposed system

The proposed system provides an integrated environment for student writing assessment rather than treating readability as a single numerical measure. After authentication, a student can enter text or upload a supported document. The application extracts readable content and presents live text statistics before the analysis is executed. The analysis result page presents the seven readability measures, text statistics, sentence information, vocabulary characteristics, grammar and spelling findings, overall writing score, and improvement suggestions.

The system separates grammar and spelling detection so that spelling mistakes can be highlighted with suggested corrections while rule-based grammar issues are displayed with explanations. Sentence analysis identifies simple, compound, and complex sentences and highlights long or very long sentences. Vocabulary analysis reports lexical richness, unique words, average word length, simple words, difficult words, and repeated words. These results are intended to help students understand the reasons behind the assessment rather than only viewing a final score.

The teacher workflow extends the same platform to assignment-based assessment. A teacher can create an assignment containing a title, instructions, minimum word count, and reference answer. Students select an available assignment, submit an answer, and receive a comparison result. The comparison uses cosine similarity between term-frequency representations of the reference and submitted answers and identifies frequently occurring reference concepts that are covered or missing. A combined result also considers the writing-quality score. Teachers can view submissions belonging to their own assignments.

The system stores each user's analysis history separately and limits teacher submission views to assignments created by that teacher. This organization provides a practical academic workflow in which writing practice, automated analysis, assignment submission, and feedback are available through one application.

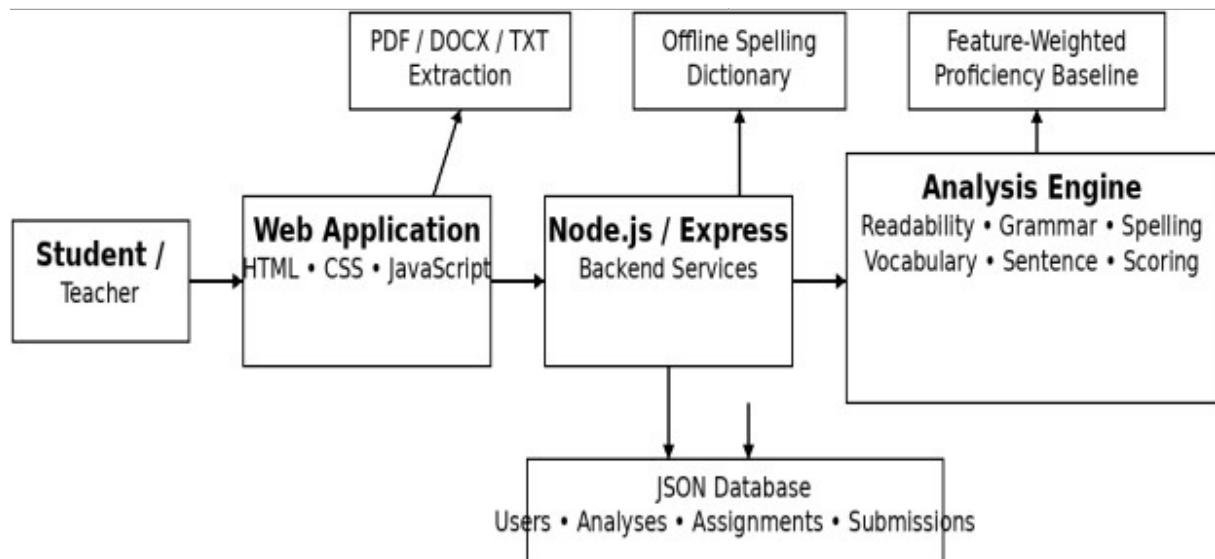


Figure 3. System architecture of the implemented web application.

6 Existing system

Conventional student writing evaluation is commonly performed through manual checking by teachers. In such a process, the evaluator reads each response, identifies errors, judges readability and vocabulary, and provides a mark or comments. Manual assessment can provide useful human judgment, but repeated checking of large numbers of submissions may require considerable effort. The level of detail in feedback can also vary between submissions.

Basic digital writing tools may provide isolated grammar or spelling checks, while readability calculators generally concentrate on one or more difficulty formulas. Such tools may not combine readability with vocabulary diversity, sentence structure, writing-quality scoring, proficiency estimation, assignment management, and personalized suggestions in one academic workflow. A separate reference-answer comparison facility may also be required when the task involves checking whether important concepts from a teacher-provided answer are covered.

The proposed system addresses these gaps by combining multiple text-analysis measures with writing-quality assessment and feedback. It does not replace teacher judgment; instead, it provides measurable indicators and stored results that can support student practice and assignment review.

Table 2. Comparison between conventional/basic assessment and the proposed system.

Feature	Conventional / Basic Tools	Proposed System
Assessment	Manual or isolated checks	Integrated writing analysis
Readability	One or limited formulas	Seven readability measures
Language Checks	Grammar or spelling separately	Grammar and spelling in one workflow
Vocabulary / Sentence	Often not combined	Richness and sentence analysis
Feedback	General or manual comments	Personalized rule-based suggestions
Assignments	External/manual process	Reference-answer assignments and submissions
History	May require separate records	User-specific analysis history
Proficiency	Not necessarily provided	Transparent feature-weighted baseline

The proposed system therefore focuses on combining several interpretable writing measures within one student-oriented application. The design is intended to support automated assistance while retaining teacher involvement for assignment creation and academic review.

7 Implementation

The prototype is implemented as a local web application using HTML, CSS, and JavaScript on the client side and Node.js with Express on the server side. The application uses JSON Web Tokens for authenticated sessions and bcryptjs for password hashing. Multer handles in-memory document uploads, pdf-parse extracts text from PDF files, and Mammoth extracts raw text from DOCX files. TXT files are read directly from the uploaded buffer.

The project uses a lightweight JSON-backed database containing user, analysis, assignment, and submission records. This design is intended for simple localhost deployment and demonstration. The server exposes separate endpoints for authentication, document extraction, spelling checks, proficiency prediction, analysis history, assignments, and submissions.

The feature-weighted proficiency component is implemented as a transparent local baseline. It normalizes the selected writing features, applies predefined positive or negative contributions, and maps the resulting score to Beginner, Elementary, Intermediate, Upper Intermediate, or Advanced. Because the current project does not include a controlled labelled-data training and validation experiment, the implementation does not report predictive accuracy as a research result.

The main implementation sequence is registration/login → writing input or document extraction → text statistics → readability and language analysis → writing-quality scoring → proficiency estimation → personalized feedback → result storage. For assignments, the sequence is teacher assignment creation → student answer submission → reference-answer similarity and concept comparison → combined assessment → submission storage.

8 Security and privacy

Security and privacy in the implemented prototype begin with authenticated access to the application. User passwords are stored as bcrypt hashes rather than plain text, and authenticated API requests use JSON Web Tokens. Role checks restrict teacher-only assignment creation and teacher submission viewing, while student submission endpoints require a student role.

Analysis history is filtered using the authenticated user's identifier, so the normal history endpoint returns only that user's saved analyses. Teacher submission views are similarly restricted to submissions associated with assignments created by the logged-in teacher. Uploaded documents are processed in memory for text extraction and are not required to be stored as original files by the analysis endpoint.

The current implementation is a localhost academic prototype and therefore does not provide the security guarantees of a production educational platform. The default JWT secret is intended to be replaced through environment configuration, and the JSON database should be replaced by a managed database for deployment. Additional controls such as HTTPS, stronger secret management, access auditing, database encryption, rate limiting, and formal security testing would be required for production use.

9 Personalized feedback and explainable analysis

Personalized feedback in the implemented system is generated from the characteristics detected during analysis. The system identifies strengths when selected measures meet defined conditions and creates improvement suggestions when weaknesses are detected. For example, a lower readability value can lead to a suggestion to simplify long or complex sentences, while low vocabulary richness can lead to advice to reduce repeated words and vary word choice. Multiple detected grammar or spelling issues can also trigger a targeted review suggestion.

The explainable analysis panel provides a short description of the factors contributing to the result. It reports the grammar and mechanics contribution, readability and average sentence length, vocabulary richness and difficult-word information, and sentence-structure score. The proficiency component similarly exposes feature contributions rather than presenting only a hidden prediction. This makes the assessment easier to interpret and gives students specific areas to work on.

10 Overall workflow

The complete workflow begins with user registration or login. After authentication, a student can enter writing directly or upload a PDF, DOCX, or TXT document. The system extracts readable text and calculates basic statistics before running the analysis modules. Readability, grammar, spelling, vocabulary, and sentence characteristics are then evaluated and passed to the writing-quality and proficiency components.

The resulting scores, detected issues, explanations, and suggestions are displayed to the student and the analysis record is stored under the authenticated user. In the assignment workflow, the teacher first creates an assignment and provides a reference answer. The student submits an answer, after which term-frequency cosine similarity and key-concept coverage are calculated and combined with writing-quality information. Submission records are then available to the appropriate student and teacher views.

This workflow connects authentication, document processing, linguistic analysis, scoring, feedback, assignment comparison, and database storage into a single application. The modular organization also allows individual analysis components to be improved without redesigning the complete system.

11 Results and discussion

The implemented prototype was exercised through the main authentication, writing-analysis, assignment, and submission workflows. The login and registration functions support separate student and teacher accounts. The writing-analysis interface accepts direct text and supported document uploads and produces a structured result containing readability measures, text statistics, sentence analysis, vocabulary information, grammar findings, spelling findings, writing-quality scoring, proficiency estimation, and improvement suggestions.

The analysis is designed to be explainable at the feature level. For example, the result can indicate the readability value, average sentence length, vocabulary richness, number of detected grammar or spelling issues, and sentence-structure contribution. Personalized suggestions are generated from these observed conditions, such as reviewing flagged issues, shortening long sentences, reducing repeated words, or improving readability through simpler wording.

The assignment workflow was also implemented with teacher-created reference answers and student submissions. The comparison returns a percentage based on term-frequency cosine similarity, covered concepts, missing concepts, word count, and a combined comparison result. These functions demonstrate how automated writing analysis can be connected with classroom-style assignments.

The present evaluation is functional rather than a controlled experimental benchmark. Therefore, the paper does not report unsupported accuracy, precision, recall, or percentage improvement values. Such numerical evaluation would require a defined dataset, expert-labelled reference scores, a fixed test protocol, and repeated experiments.

11.1 Advantages

- Provides multiple readability measures instead of relying on a single formula.
- Combines grammar, spelling, vocabulary, sentence, and writing-quality analysis.
- Provides personalized and explainable improvement suggestions.
- Supports PDF, DOCX, and TXT document input.
- Maintains user-specific analysis history and assignment records.
- Supports teacher reference answers and student submission comparison.

11.2 Limitations

The current prototype uses rule-based grammar checking and dictionary-based spelling detection, so it cannot identify every context-dependent language error. Readability formulas are statistical indicators and should not be treated as complete measures of writing quality. The proficiency predictor is a transparent baseline and has not been validated as a research-grade model. Scanned PDFs are not supported because OCR is not included. The JSON-backed database is suitable for local demonstration but is not intended for large-scale deployment.

12 Future enhancement

Future development can extend the system with larger labelled datasets and controlled validation of the proficiency-prediction component. More advanced language models can be studied for contextual grammar correction, semantic writing assessment, and improved feedback generation. OCR can be added to support scanned documents, while stronger reference-answer comparison can use semantic embeddings instead of only term-frequency similarity.

Further enhancements can include progress graphs across multiple assessments, teacher-level analytics, class performance summaries, context-aware suggestions, plagiarism detection, mobile support, and configurable assessment rubrics. A managed database, secure deployment configuration, HTTPS, audit logging, and stronger access controls can also be introduced when moving beyond localhost use.

13 Conclusion

The NLP-Based Student Readability Assessment and Personalized Feedback System provides an integrated web-based environment for automated student writing assessment. The implemented workflow combines seven readability measures, text statistics, rule-based grammar checking, offline spelling detection, vocabulary analysis, sentence analysis, writing-quality scoring, proficiency

estimation, and personalized feedback. The system also supports document uploads and stores user-specific assessment history.

The assignment component extends the system beyond standalone text checking by allowing teachers to create reference-based assignments and students to submit answers for comparison. The prototype demonstrates how measurable writing characteristics can be presented in an explainable manner to support student improvement. The current implementation is intended as an academic project prototype; further dataset-based validation, semantic analysis, stronger language models, and production-level security would be required for broader educational deployment.

Acknowledgments

The authors acknowledge the guidance and technical support provided by KIRAN B, Assistant Professor, Department of Computer Science & Engineering, Mysuru Royal Institute of Technology, Mysuru, Karnataka, India, during the development of this project.

References

- [1] Misgna H, On BW, Lee I, Choi GS. A survey on deep learning-based automated essay scoring and feedback generation. *Artificial Intelligence Review*. 2025;58:36.
- [2] Gombert S, Fink A, Giorgashvili T, Jivet I, Di Mitri D, Yau J, et al. From the automated assessment of student essay content to highly informative feedback: A case study. *International Journal of Artificial Intelligence in Education*. 2024;34:1378–1416.
- [3] Liu Y, Han J, Sboev A, Makarov I. GEEF: A neural network model for automatic essay feedback generation by integrating writing skills assessment. *Expert Systems with Applications*. 2024;245:123043.
- [4] Goldshtein M, Alhashim AG, Roscoe RD. An NLP-based exploration of variance in student writing and syntax: Implications for automated writing evaluation. *Computers*. 2024;13(7):160.
- [5] Dai W, Tsai YS, Lin J, Aldino A, Jin H, Li T, et al. Assessing the proficiency of large language models in automatic feedback generation: An evaluation study. *Computers and Education: Artificial Intelligence*. 2024;7:100299.
- [6] Hou Z, Ciuba A, Li X. Improving LLM-based automatic essay scoring with linguistic features. In: *Proceedings of the Innovation and Responsibility in AI-Supported Education Workshop*. PMLR. 2025;273:41–65.
- [7] Zhang C. Optimising AI writing assessment using feedback and knowledge graph integration. *PeerJ Computer Science*. 2025;11:e2893.