PRESIDENCY UNIVERSITY
Private University Estd. in Karnataka State by Act No. 41 of 2013
Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

Application for Assessment of Quality of Textbook/Reference Books/E-Book

A PROJECT REPORT

Submitted by

Lavanya M- 20221CSE0392
Posina Supraja- 20221CSE0266
Talluru Chethana- 20221CSE0385

Under the guidance of,

Dr. MD ZIAUR RAHMAN

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

PRESIDENCY UNIVERSITY

BENGALURU

DECEMBER 2025

PRESIDENCY UNIVERSITY
Private University Estd. in Karnataka State by Act No. 41 of 2013
Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this report “Application for Assessment of Quality of Textbook/Reference Books/E-Book” is a Bonafide work of “Lavanya M (20221CSE0392), Posina Supraja(20221CSE0266), Talluru Chethana (20221CSE0385)”, who have successfully carried out the project work and submitted the report for partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING during 2025-26.

Dr. Md Ziaur Rahman
Project Guide
PSCS
Presidency University

Dr. Muthuraju V
Program Project
Coordinator
PSCS
Presidency University

Dr.Sampath A K
Dr,Geeta A
School Project
Coordinators
PSCS
Presidency University

Dr. Blessed Prince
Head of the
Department
PSCS
Presidency University

Dr. Shakkeera L
Associate Dean
PSCS
Presidency University

Dr. Duraipandian N
Dean
PSCS & PSIS
Presidency University

Name and Signature of the Examiners

1)

2)

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We the students of final year B.Tech in COMPUTER SCIENCE AND ENGINEERING, at Presidency University, Bengaluru, named Lavanya, Posina Supraja, T Chethana, hereby declare that the project work titled “Application for Assessment of Quality of Textbook/Reference Books/E-Book” has been independently carried out by us and submitted in partial fulfillment for the award of the degree of B.Tech in COMPUTER SCIENCE AND ENGINEERING during the academic year of 2025-26. Further, the matter embodied in the project has not been submitted previously by anybody for the award of any Degree or Diploma to any other institution.

Lavanya M	USN: 20221CSE0392
Posina Supraja	USN: 20221CSE0266
Talluru Chethana	USN: 20221CSE0385

PLACE: BENGALURU

DATE: 28-NOVEMBER 2025

ACKNOWLEDGEMENT

For completing this project work, We have received the support and the guidance from many people whom I would like to mention with deep sense of gratitude and indebtedness. We extend our gratitude to our beloved Chancellor, Pro-Vice Chancellor, and Registrar for their support and encouragement in completion of the project.

I would like to sincerely thank my internal guide **Dr. Md Ziaur Rahman**, Assistant Professor, Presidency School of Computer Science and Engineering, Presidency University, for his moral support, motivation, timely guidance and encouragement provided to us during the period of our project work.

I am also thankful to **Dr. Blessed Prince**, Professor, Head of the Department, Presidency School of Computer Science and Engineering, Presidency University, for his mentorship and encouragement.

We express our cordial thanks to **Dr. Duraipandian N**, Dean PSCS & PSIS, **Dr. Shakkeera L**, Associate Dean, Presidency School of Computer Science and Engineering, and the Management of Presidency University for providing the required facilities and intellectually stimulating environment that aided in the completion of my project work.

We are grateful to **Dr. Sampath A K**, and **Dr. Geetha A**, PSCS Project Coordinators, **Dr.Muthuraju V**, Program Project Coordinator, Presidency School of Computer Science and Engineering, for facilitating problem statements, coordinating reviews, monitoring progress, and providing their valuable support and guidance.

We are also grateful to the Teaching and Non-Teaching staff of Presidency School of Computer Science and Engineering, and also staff from other departments who have extended their valuable help and cooperation throughout the course of this project.

LAVANYA M

POSINA SUPRAJA

TALLURU CHETHANA

ABSTRACT

Textbooks continue to serve as essential pedagogical resources in higher education, yet the assessment of their quality often remains fragmented, subjective, and limited in scope. Scholarly studies by Cobb (1987), Gurung and Martin (2011), Fey and Matthes (2018), and others emphasize the need for structured frameworks, user-centered evaluation, and transparent quality criteria to ensure that textbooks align with curricular and pedagogical goals. Building on these insights, this project, titled **"Application for Assessment of Quality of Textbook/Reference Books/E-Book (TextAssess)"**, presents a full-stack web application that operationalizes these research-based evaluation principles within a digital ecosystem.

The frontend of TextAssess utilizes React.js. The system also uses TypeScript, and even Tailwind CSS; This ensures a very intuitive, very responsive user interface. The interface provides textbook browsing capabilities. One can also submit reviews! Then there are the academic discussion elements, plus user dashboards. Even the administrative aspect can be monitored. The backend incorporates Supabase to do even better, built on PostgreSQL of course! It provides safe data storage. It ensures safe user authentication and access. The system allows role-based permissions; a plus for the students and administrators. Evaluation can become quite participatory in nature because of this design. Users give qualitative or quantitative assessments, all centered on clarity. Relevance and accessibility matters too. They also value, of course, good pedagogical content. All taken from past writings and research

The system underwent some crucial testing. Testing involved various sample datasets. Testing has demonstrated clear synchronization; seamless between the frontend and backend really. One can efficiently find user-written reviews; and have evaluation metrics measured correctly too! Admins have review-approval access now and can overview feedback. So quality summaries can be made, also very important to consider. Such things promote great selections of resources. Messaging and threaded discussion exist as collaborative aspects. They help people work together smoothly. These boost transparency and help people truly engage.

Ultimately, TextAssess links the abstract; textbook evaluation with true data analysis. A robust framework evaluation comes through with adaptable scalable designs. That really sums the system. This presents one solution regarding institutional evaluations; Textbook assessment fosters more transparency and also progress. Overall it makes a difference for those needing better educational selections.

TABLE OF CONTENT

List of Figures

Abbreviations

Abbreviation	**Full Form**
API	Application Programming Interface
SPA	Single Page Application
CSR	Client-Side Rendering
DBMS	Database Management System
PGSQL / PostgreSQL	PostgreSQL Database
SDK	Software Development Kit
NPM	Node Package Manager
RDBMS	Relational Database Management System
TLS	Transport Layer Security
SSL	Secure Sockets Layer
AES	Advanced Encryption Standard
OTP	One-Time Password
ACL	Access Control List
TSX	TypeScript XML (React Component File)
RBA	Role-Based Access
UIM	User Interface Module
RBA	Role-Based Access
FRD	Functional Requirement Document
MSM	Messaging Module

Chapter 1

INTRODUCTION

Textbooks are very important; they are central to the teaching–learning process and shape curriculum delivery as well as the quality and depth of learner engagement. Their effectiveness depends on conceptual clarity, pedagogical design, content relevance, accessibility, and alignment with educational standards. However, traditional textbook evaluation done by small reviewer panels or individual educators faces challenges like subjectivity, limited transparency, inconsistent review criteria, and lack of student perspective.

To address these issues, this project presents TextAssess, an intelligent, web-based platform designed for collaborative and data-driven evaluation of textbooks, reference books, and e-books. TextAssess encourages active participation from professors, students, reviewers, and administrators, offering a structured interface for review submission, quality rating, and academic discussions. It also generates institutional insights for textbook selection and policy-making, recognizing that textbook quality has long-term impact.

Unlike traditional methods, TextAssess leverages full-stack technologies to ensure an effective, scalable, and user-friendly evaluation process. The frontend is built using React.js, TypeScript, and Tailwind CSS, providing modules for book exploration, review submission, and discussion threads, along with interactive dashboards for different user roles. The backend, powered by Supabase (a cloud-based service built on PostgreSQL), manages authentication, secure data storage, and real-time synchronization. Role-based access control enables administrators to oversee approvals while allowing students and reviewers to contribute under defined privileges.

The project aligns with global efforts in educational quality assurance. Reports from UNESCO and national education boards highlight continued use of outdated learning materials, which hinders learning outcomes. Existing digital libraries and e-learning platforms often lack research-based textbook evaluation mechanisms. TextAssess fills this evaluation gap by offering a transparent, collaborative, and data-backed solution, effectively digitalizing textbook assessment frameworks.

Through the use of cloud technologies, TextAssess transforms fragmented manual evaluation into a systematic, participatory, and data-driven process. This supports evidence-based textbook selection and contributes to educational improvement at institutional, state, and national levels truly realizing the vision of "Better Textbooks for Better Learning."

1.1 Background

Textbooks remain crucial for education. They are the essential tool for structured learning and instruction that is guided. Their quality shapes comprehension. Textbook quality impacts curriculum alignment. The quality, therefore, affects student performance overall. High-quality textbooks, carefully designed and thoroughly evaluated, improve educational outcomes. Clear explanations, well-organized content, and helpful exercises make the improvement. This is according to research [1].

But, evaluating textbooks is still a traditional process. Manual reviews take place sometimes. Evaluation is often subjective and fragmented which is not a positive thing. Review committees or individual educators look at learning material, but employ varying standards which does limit the transparency. In addition stakeholder participation does have an effect on textbooks being produced. Research suggests a collaborative approach to the standardization, proves very important for assuring that textbooks meet both learner and pedagogical needs [2][3]. I actually feel research should have a larger part.

Digital learning platforms grow these days and the same goes for open educational resources. Both make educational content far easier to get; in other words more accessible. Yet these platforms sometimes are not doing well on the mechanisms needed for proper evaluation of textbook quality, of course. Global studies, reveal that around the world; in developing regions; more than one third; roughly up to 35 percent; lack high-quality, current textbooks; these inequalities certainly affects learning outcomes [4]. Further; institutional textbook choices seem to avoid getting student insights or information, and sometimes data from good old data driven evaluation [5].

We propose **TextAssess** to close some of these gaps, though other options could be considered, that will be another problem to address, by the way. TextAssess, is, or rather it would be a cloud-based, web app intended to foster cooperation, in other words is designed for cooperative evaluation. Ratings are assessed, along with discussion, centrally; between the parties: student, professors, administrators, of course, reviewers, too, it includes a bit of every person important to textbooks being implemented or denied. **React.js and TypeScript** are included in developing along with **Tailwind CSS**, which is on the front. The backend is **Supabase** (PostgreSQL, Authentication, Storage), because we wanted a scalable, transparent and solution; what's even better efficient, even when discussing relevance, as we will use this system of evaluation. This helps with getting the quality of educational texts at academic institutes.

1.2 Statistics

Textbooks are crucial for a good education. But some schools use old or bad books. Worldwide, nearly 250 million students do not have good, current textbooks. This causes big gaps in basic reading and math skills[4]. New studies also tell us something interesting. Good books and good student performance do in fact go together; this is particularly important in poorer countries.

In India, surveys find that most college students; over **70 percent**, use textbooks that are not up-to-date, maybe badly matched to the class. Lots of colleges also do not have strong ways to judge textbooks. The issue impacts what a student really knows and if they are ready for jobs[6].

New studies on free educational resources online or **OER**s are interesting, they appear hopeful; in both ease of access and how affordable they can be. Sadly, one major weakness remains to be the lack of formal quality-checking systems for it all. However, evidence does show a possibility. Platforms using group checking and teamwork get far more reliable, even very good results.

TextAssess wants to fix this, It obtains feedback from faculty, reviewers, along with their respective students. This uses both rating numbers combined together alongside thoughtful written insights from each contributor, it will close this hole, I am convinced. TextAssess changes judging things like that into a visible process, as this all uses factual, supporting, strong evidence, the platform seeks, through every thing, to offer the best. That is that schools make the smartest decision selecting the greatest textbooks. Textbooks that adhere both to lesson goals or very best new, modernized ways of instruction that's available [10].

1.3 Prior existing technologies

The evaluation of textbooks has historically depended on **manual review checklists** designed by curriculum boards or academic committees. While these systems ensured a level of expert participation, they often lacked flexibility, transparency, and the inclusion of learner perspectives [1]. Early frameworks primarily assessed surface-level attributes such as factual accuracy, topic sequence, and alignment with the syllabus but rarely incorporated feedback from the actual users — students and instructors.

Over time, researchers proposed **structured evaluation models** to introduce systematic quality assurance. Studies emphasized the need to assess not only content accuracy but also pedagogical soundness, readability, and cultural appropriateness [3][5]. For instance, some national education systems adopted rubric-based frameworks, yet these remained static and paper-based, leading to inconsistencies across institutions.

Empirical and theoretical work expanded the discussion on textbook quality. Comparative analyses of multiple textbooks [1] showed wide variation in clarity, exercises, and teaching alignment, while psychometric studies [2] demonstrated that students' perceptions of textbook usability directly influence their reading behavior and exam performance. Research also identified critical gaps in existing evaluation processes, particularly the lack of standardized digital mechanisms that allow collective participation [4][5].

International and national reviews further revealed that institutional textbook approvals often fail to meet expected quality benchmarks [5][7]. The absence of learner input and evidence-based metrics leads to the continuous adoption of materials that may not meet pedagogical needs [8]. Moreover, studies on textbook quality standards [7] and annotated bibliographies [6] have highlighted that evaluation should move beyond content coverage to consider multimedia design, inclusivity, and interactivity—elements increasingly important in modern digital learning environments.

The emergence of **Open Educational Resources (OERs)** transformed access to learning content but introduced new challenges. While OER platforms improved affordability and reach, they rarely implemented structured quality control frameworks [9]. Research on OER adoption [9][10] found that students benefit from open-access materials but that the absence of systematic peer review affects consistency and reliability. In addition, most existing digital repositories such as MERLOT, OpenStax, or academic forums focus on content distribution rather than verified academic evaluation [7]. Similarly, commercial websites and e-learning marketplaces offer user-generated reviews that, while useful, are unverified and lack the methodological rigor required for academic decision-making.

Earlier technological attempts at automating evaluation used **static web technologies** HTML, CSS, JavaScript, and localStorage—to demonstrate feasibility for collecting reviews and ratings. These prototypes [4] proved that digital feedback loops could enhance transparency but suffered from limited scalability, absence of secure data handling, and lack of real-time synchronization.

Drawing on these findings, **TextAssess** was conceptualized to bridge the gap between **research-based frameworks and technological implementation**. The system incorporates principles from comprehensive evaluation models [1][3][5][6][7] and empirical insights on user perception and feedback [2][4][8][9][10], transforming them into a live, data-driven platform.

- The **frontend**, built using **React.js, TypeScript, Tailwind CSS, and shadcn/UI**, provides a dynamic, modular interface where users can browse, submit, and review textbooks interactively.
- The **backend**, powered by **Supabase**, offers secure PostgreSQL-based data storage, real-time updates, and role-based authentication for students, professors, reviewers, and administrators.

By integrating research-backed evaluation criteria [1] through [10] into a modern cloud architecture, **TextAssess** ensures consistent, transparent, and collaborative textbook assessment. The platform not only digitizes prior evaluation frameworks but also operationalizes them—transforming manual, periodic reviews into a continuous, community-driven quality assurance process that aligns with 21st-century educational needs.

1.4 Proposed approach

- **Aim of project :**

  The primary aim of this project is to design and implement a **web-based platform (TextAssess)** that enables professors, reviewers, and students to collaboratively evaluate textbooks. The platform seeks to improve the transparency, accessibility, and reliability of textbook reviews by combining structured input forms, rating systems, and a communication channel for clarifications.

- **Motivation:**

  Textbooks continue to play a very big part in learning. However, the textbook quality can greatly differ among institutions. Traditional review committees sometimes focus on brief checklists; rarely capturing what the students really think [1][2][5]. As a resul

1.5 Objectives

- **Behaviour**

  To design and implement an interactive user interface that supports seamless navigation between modules such as Home, Books, Submit, Profile, and Messages, enabling role-based actions (student, professor, reviewer, admin) within the system.

- **Analysis**

  To develop functionality for collecting, filtering, and analyzing book-related data such as average ratings, number of reviews, and pending approvals—providing structured insights into textbook quality.

- **System Management**

  To create an **Admin Dashboard** that manages submitted books, monitors reviews, and controls approval workflows, ensuring effective moderation and organized system operation.

- **Security**

  To incorporate a **role-based login and signup mechanism** that restricts unauthorized access, ensuring that only registered users can submit books, write reviews, or send messages.

- **Deployment**

  It needs to develop a prototype, and we will host it on React using the **Supabase** prototype for sure. The prototype can then shows collaboration, is scalable, provides cloud based integration so it's great to offer the bedrock to improve it. AI and analytics enhancement and that will be the future.

Chapter 2

LITERATURE REVIEW

[1] Cobb, George W. - Introductory Textbooks: A Framework for Evaluation: A Comparison of 16 Books.

Cobb's study was one of the earliest efforts to create a clear framework for evaluating textbooks. He compared sixteen introductory statistics books and identified significant factors that affect textbook quality, such as structure, clarity, examples, and overall organization. Cobb argued that a good textbook should convey content and make it understandable through logical sequencing and relatable examples.

Limitation: The study was limited to statistics and did not test the framework in other subjects. Relevance: Cobb’s structured approach to comparing multiple textbooks helped shape the evaluation rubrics used in TextAssess, ensuring that reviews cover both teaching quality and ease of understanding.

[2] Gurung, Regan A. R. & Martin, Ryan C. - Predicting Textbook Reading: The Textbook Assessment and Usage Scale

Gurung and Martin focused on how students use their textbooks. They developed the Textbook Assessment and Usage Scale to understand which features make students more likely to read and engage with their course materials. They found that clarity, visual aids, and logical organization significantly influenced student reading habits and overall performance.

Limitation: The study relied on self-reported data from students, which may include bias. Relevance: Their findings highlight the need for student involvement in evaluation, which inspired TextAssess to include student feedback options alongside instructor and reviewer evaluations.

[3] Fey, Carl-Christian & Matthes, Eva - Textbook Quality Criteria and Evaluation

Fey and Matthes examined textbook quality from a broader academic perspective. They discussed how educational materials should be assessed not only for factual accuracy but also for cultural balance, inclusivity, and support for different teaching methods. Their work proposed that quality should be viewed as multi-dimensional, combining content, teaching methods, and design.

Limitation: The study is largely theoretical and lacks direct experimental testing.

Relevance: Their multi-dimensional model supports TextAssess’s rubric-based design, which evaluates textbooks on various factors like content quality, structure, inclusivity, and accessibility.

[4] Mohammadi, Mohammad & Abdi, Heidar - Textbook Evaluation: A Case Study This study presented a practical evaluation of an English language textbook used in Iranian universities. Both students and teachers participated in the assessment, revealing that while the textbook was linguistically sound, it lacked interactive and practical exercises.

Limitation: The study was context-specific and limited to English language teaching. Relevance: The inclusion of both teacher and student perspectives in their evaluation reinforced the idea of multi-stakeholder feedback, which became a key principle in the design of TextAssess.

[5] Mahmood, Khalid - Conformity to Quality Characteristics of Textbooks: The Illusion of Textbook Evaluation in Pakistan

Mahmood's research critically analyzed Pakistan's government-approved textbooks and found that many did not meet the expected quality standards, despite being approved for official use. The study pointed out that textbook reviews were often superficial and lacked transparency.

Limitation: The study mainly captured expert opinions and excluded student feedback.

Relevance: Mahmood's work emphasizes the need for transparent, digital evaluation systems, directly supporting TextAssess's goal of making textbook reviews open, trackable, and based on data.

[6] Woodward, Arthur; Elliot, David L.; & Nagel, Kathleen C. - Textbooks in School and Society: An Annotated Bibliography & Guide to Research

Woodward and colleagues compiled a guide to research on how textbooks influence both classroom learning and broader social values. They argued that textbooks are not just educational tools but also social instruments that shape perspectives and knowledge.

Limitation: The publication serves as a research guide rather than a direct evaluation framework.

Relevance: Their work encouraged the inclusion of cultural and ethical dimensions in TextAssess, ensuring that reviews cover not just content accuracy but also diversity, representation, and checks for bias.

[7] Ivić, Ivan; Antić, Slobodanka; & Pešikan, Ana - Textbook Quality: A Guide to Textbook Standards

This publication set out clear standards for textbook development and evaluation. It emphasized aspects such as curriculum alignment, visual presentation, inclusivity, and the need for textbooks to support teacher guidance.

Limitation: The proposed standards are general and may need adjustment for specific subjects.

Relevance: These guidelines shaped the evaluation criteria within TextAssess, ensuring consistency across disciplines while allowing for customization.

[8] Yasar, Okan - A Comparative Analysis of Assessment and Evaluation Exercises in Geography Textbooks

Yasar analyzed geography textbooks under two different Turkish curricula and found that newer editions encouraged critical thinking and active learning more effectively than older versions. The study highlighted how curriculum design affects the quality and relevance of textbook activities.

Limitation: The findings are limited to one subject area.

Relevance: This study supports TextAssess's comparative analytics feature, allowing users to compare different textbooks or editions and identify which best meets evolving curriculum standards.

[9] Race, Phil - The Lecturer's Toolkit: A Practical Guide to Assessment, Learning and Teaching

Race discussed how teaching methods and textbook usage go hand in hand. He argued that textbooks are only truly effective when paired with proper instructional strategies, assessment design, and learner engagement techniques.

Limitation: The book is practical rather than research-based, offering general teaching advice.

Relevance: Race's insights reinforced the idea that TextAssess should connect textbook evaluation with teaching practice, helping educators select books that complement their instructional goals.

[10] Giger, Joyce Newman & Haddad, Linda - Transcultural Nursing: Assessment and Intervention

Although primarily a nursing textbook, Giger and Haddad's work shows how textbooks can integrate real-world applications, cultural sensitivity, and contextual learning. It serves as an example of how discipline-specific materials can address both knowledge and empathy.

Limitation: It is not a research study, so it lacks analytical evaluation data.

Relevance: This book inspired TextAssess's subject-specific flexibility, where evaluation rubrics can adapt to different fields, ensuring that discipline-specific needs are met.

Chapter 3

METHODOLOGY

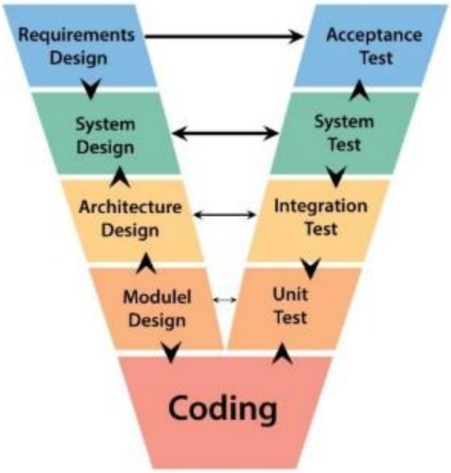


Fig 3.1 The V model methodology

The **V-Model** is a software development lifecycle that highlights a parallel relationship between each development activity (on the left side) and its corresponding verification and validation phase (on the right side).

For the **TextAssess** application, this approach ensures that every requirement is carefully mapped through design, implementation, and testing so the final platform effectively supports educators and students in assessing textbook quality [1][2][3].

3.1 Requirement Analysis and Specification

Activities:

- Gathered stakeholder inputs from educators, students, and reviewers.
- Defined evaluation parameters: *pedagogical accuracy*, *curriculum alignment*, *clarity*, *examples*, and *inclusivity*.
- Outlined backend requirements such as authentication, user accounts, book uploads, review moderation, and admin-level reporting.
- Established data privacy guidelines — only emails are visible for contact; phone numbers are hidden.

Rationale**:**

High-quality textbooks directly influence learning outcomes. Explicitly defined requirements ensure that evaluation parameters translate into measurable indicators of quality [1][2][4].

Deliverables:

- Requirement documentation
- Acceptance criteria
- Data privacy and storage policy
- Standardized evaluation rubrics (for internal and external assessment) [7][10] **3.3 System Design Phase**

Activities:

- Defined the full-stack architecture using **React.js + TypeScript** for the frontend and **Supabase (PostgreSQL, Auth, Storage)** for the backend.

- Established a secure communication channel between frontend and backend using Supabase APIs.
- Designed data models for users, books, reviews, and messages.
- Integrated analytics modules for computing ratings and visualizing data on the admin dashboard.
- Defined security rules such as JWT-based role control and restricted visibility of personal data.

Rationale:
The choice of a **Supabase cloud backend** ensures real-time synchronization, while research highlights how access to well-structured content improves learning and scalability [4][11].

Deliverables:
System architecture diagram, API documentation, database schema, and analytics integration points.

3.2 Architectural Design and Technology Selection

Activities:

- Selected **React.js** with Tailwind CSS and shadcn/UI for a clean, accessible frontend interface.
- Used **Supabase** for backend authentication, real-time database operations, and cloud file storage.
- Defined authentication and role-based access workflows.
- Created an automated review and moderation flow for administrators.
- Designed the reporting module to summarize user activity and textbook statistics.

Rationale:
Studies on Open Educational Resources (OER) [5][9] indicate that systems should accommodate both open-access and restricted textbooks. Hence, the architecture was designed to support flexible content access and analytics for quality tracking.

Deliverables:
Component diagrams, integration testing plans, and data migration strategy.

3.3 Module Design and User Interface Flow

Activities:

- Developed responsive UI modules using React components: *Home, Books, Submit, Reviews, Profile, and Messages.*
- Backend services included *Book Uploads, Review Moderation, Messaging System,* and *Analytics Generator*.
- Built the Admin Dashboard for overseeing submissions and review approvals.
- Defined usability standards ensuring accessibility and smooth navigation.

Rationale:
Incorporating user-centered design ensures inclusivity and ease of use. Prior studies emphasize the need to include *student voice* and *learner feedback* during interface design [6].

Deliverables:
Wireframes, component specifications, and unit test cases.

3.4 Coding and Implementation

Activities:

- Implemented frontend using **React.js + TypeScript** with Tailwind CSS for styling.

- Developed backend logic using **Supabase APIs** for authentication, CRUD operations, and file storage.
- Integrated **React Query** for data fetching and real-time updates.
- Ensured all private user data (phone numbers) remained hidden from public access.
- Linked all major workflows: book upload → review → approval → analytics.

Deliverables:
Source code repository, documentation, and unit test results.

3.5 Unit Testing (Component-Level Validation)

Tests:

- Validated form inputs and error handling on book submission and review pages.

- Verified file uploads and metadata retrieval from Supabase Storage.
- Tested CRUD operations for reviews and messages.
- Checked privacy enforcement (email-only visibility).

Goal:
To confirm each component or service functions correctly before integration.

3.6 Integration Testing (Frontend–Backend Connectivity) Tests:

- Verified frontend communication with Supabase APIs.
- Tested authentication across all roles (student, professor, admin).
- Ensured proper linking of reviews with book entries.
- Checked data flow consistency between UI, database, and analytics.

Goal:
To validate that modules interact seamlessly and maintain synchronized, error-free data.

3.7 System Testing (End-to-End Validation)

Tests:

- Conducted complete workflow checks: textbook submission, review submission, admin approval, and analytics visualization.
- Performed load and concurrency testing for multiple simultaneous submissions.
- Ensured secure access control and non-disclosure of sensitive data.

Goal:
To ensure that the system satisfies both functional and non-functional requirements such as usability, security, and performance.

3.8 Acceptance Testing (Stakeholder Validation)

Tests:

- Educators, students, and reviewers tested the system using real data.
- Verified that book details displayed correctly, analytics matched reviews, and privacy measures were followed.

- Ensured compliance with evaluation rubrics and role-based privileges [7][8][10].

Goal:

To validate that **TextAssess** meets user needs and performs as an effective academic tool for textbook quality assessment.

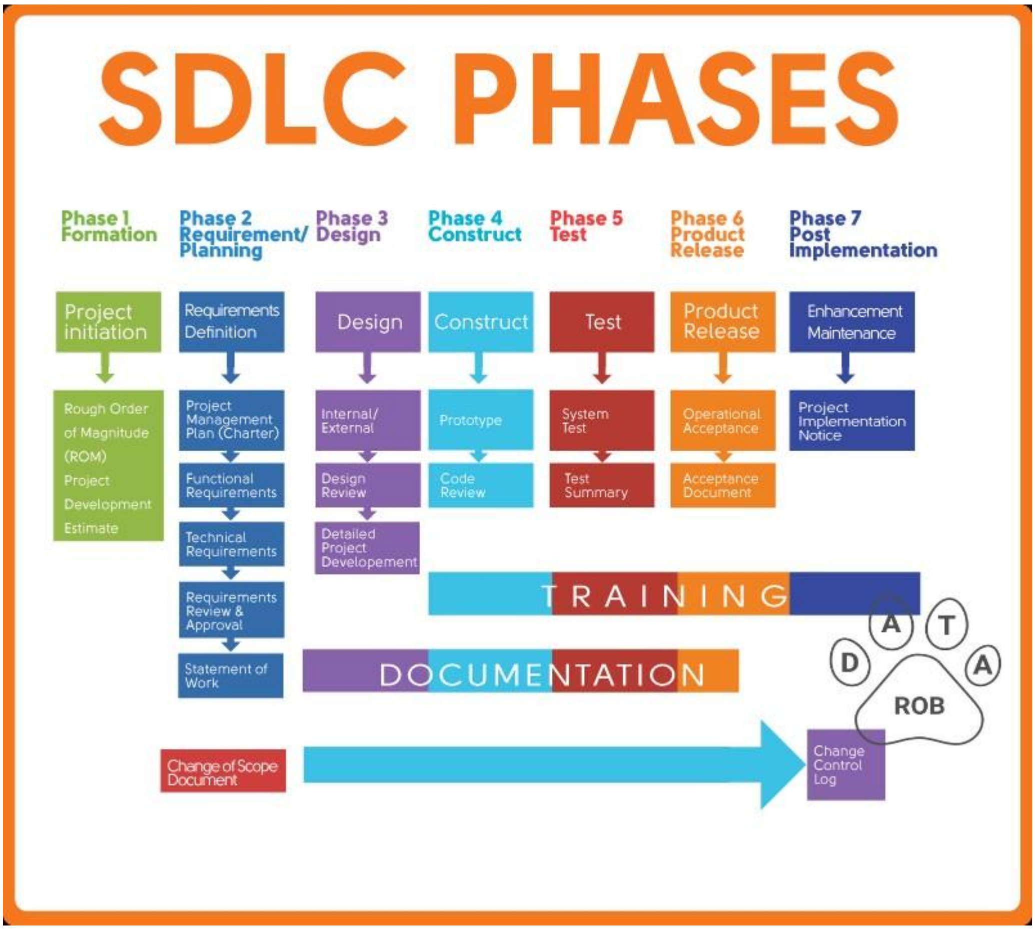


Fig 3.2 SDLC phases

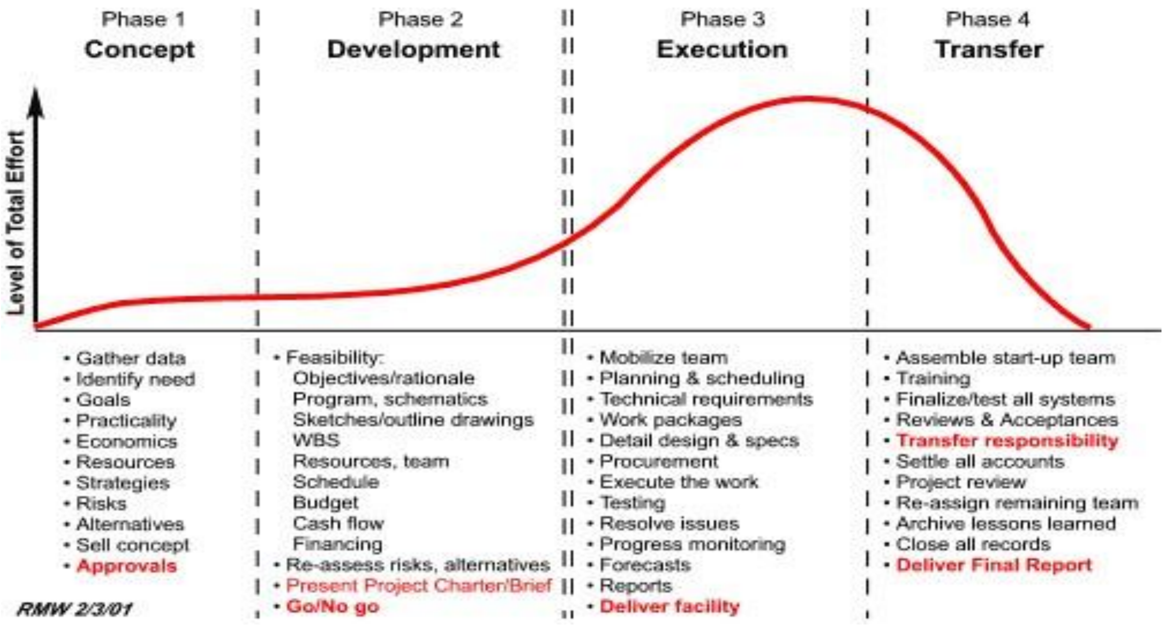


Fig 3.3 Summary of project breakdown to task

Chapter 4

ANALYSIS AND DESIGN

4.1 Requirements

The **Analysis and Design phase** is where ideas start turning into structure. This stage acts as a bridge between understanding the problem and creating a working system. In this project, the analysis focuses on *what* the system must accomplish — a reliable, easy-to-use, and transparent platform for textbook evaluation — while the design outlines *how* these goals are implemented using the chosen technologies and frameworks [1][2].

This phase draws on the principles highlighted in earlier research, ensuring that every design choice is grounded in proven approaches to textbook evaluation, user experience, and educational technology [3][5][7][10].

Hardware Requirements

Even though **TextAssess** is a web-based platform, some basic hardware and hosting conditions are essential to keep the system stable and responsive.

- **Initial Conditions:**
  The platform is designed to run on any modern computer or mobile device using browsers like Chrome, Edge, or Firefox. During development, a standard system capable of running Node.js and development tools is required.
- **Input Parameters:**
  Users will upload textbooks (or their details), fill out review forms, and interact through dashboards. These actions demand stable internet and sufficient server resources for smooth uploads and quick responses.
- **System Outcomes:**
  The hardware setup should handle user traffic, store and retrieve data efficiently, and display textbook details, ratings, and analytics without lag.

- **Constraints:**
  Since the platform uses **cloud hosting through Supabase**, there's no need for heavy on-premise hardware. However, internet connectivity is essential, which can be a limitation for some users — a challenge also noted in previous studies on digital textbook adoption [4][11].

Software Requirements

The **software design** defines the technologies and frameworks used to bring the system to life. The platform is built as a **full-stack web application**, combining a strong, interactive frontend with a secure, cloud-backed database.

Frontend (User Interface Layer):

- **Technologies Used:**
  React.js with TypeScript ensures modular, efficient, and error-free code. Tailwind CSS and shadcn/UI provide a clean, modern, and responsive design across devices. React Router manages navigation, while React Query handles real-time data fetching and caching.
- **Purpose:**
  The frontend focuses on simplicity and accessibility, allowing students, professors, and reviewers to navigate easily and submit reviews without confusion.
- **Literature Link:**

Studies have shown that user-centered design and intuitive interfaces increase engagement and accuracy in educational systems [2][6][9].

Backend (Server and Database Layer):

- **Technologies Used:**
  Supabase powers the backend with its integrated **PostgreSQL database, authentication module, and cloud storage**. Supabase Auth handles role-based access control (students, professors, reviewers, admins), while the real-time database manages all interactions and submissions.
- **Purpose:**
  To store and manage user profiles, textbook metadata, reviews, messages, and analytics securely.
- **Literature Link:**
  Cloud-based solutions like Supabase align with modern OER (Open Educational Resource) frameworks that emphasize scalability and accessibility [5][9].

- **Constraints:**
  The system depends on Supabase's cloud services, meaning performance depends on plan limits and internet reliability. However, this also reduces maintenance costs and increases portability — a balance suggested in textbook digitization studies [3][8].

Data Collection Requirements

Data is the foundation of TextAssess. To make evaluations meaningful and comparable, the system collects structured and standardized information.

- **Collected Data:**
  Textbook details (title, author, edition, subject, and summary), user roles, structured review data (rubric-based ratings, comments), and optional uploaded textbook files.
- **Consistency and Validation:**
  Review fields follow standardized rubrics inspired by academic evaluation frameworks [1][3][7]. Each review includes quantitative scores and qualitative feedback, ensuring a balanced and fair assessment.
- **Privacy:**
  Reviewer identities are partly masked — only email contact is visible — ensuring open participation while maintaining accountability [10][11].
- **Workflow:**
  1. Submitter uploads textbook information or files.

2. Reviewer fills in rubric-based evaluations.
3. Admin moderates reviews for approval ● **Link to Research:**

This flow mirrors real-world academic review processes discussed in prior literature, promoting fairness, consistency, and transparency [5][7][10].

Data Analysis Requirements

Collected data is analyzed to generate useful insights for users and administrators.

- **Core Functions:**
  - ○ Calculate average ratings and quality scores.
  - ○ Display trends through visual dashboards.
  - ○ Compare different textbooks or editions based on ratings, criteria, or subject areas.

- **Advanced Analysis:**

  The system can detect patterns, such as frequently mentioned issues (e.g., "outdated content" or "poor examples"), and highlight the most recommended books.

- **Validation:**

  Each review follows an evaluation rubric ensuring fair comparison, consistent with global textbook evaluation standards [2][7][9].

- **Outcome:**

  These analytics help educators, students, and decision-makers choose textbooks based on real, collective feedback rather than personal bias [3][5].

System Management Requirements

System management features ensure the platform remains reliable, organized, and moderated.

- **Roles and Permissions:**

  Users are divided into four main roles: Student, Professor, Reviewer, and Admin each with specific privileges.
- **Admin Dashboard:**

  Allows administrators to approve or reject reviews, manage users, and oversee analytics reports.

- **Moderation:**
  Built-in mechanisms prevent biased or spam reviews, with admins authorized to flag or remove inappropriate submissions.

Version Tracking:
The system records book editions and keeps previous review data for reference, a practice supported by comparative evaluation research [8][10].

Security Requirements

Security is a top priority for TextAssess, especially since user data and reviews are stored in the cloud.

- **Authentication:**
  Implemented through **Supabase Auth** using JWT-based tokens. Each login is role-protected, ensuring that only authorized users can access or modify specific data. ● **Encryption:**
  All data transfers use HTTPS to protect user credentials and sensitive book metadata.
- **Privacy Controls:**
  User phone numbers remain hidden; only verified emails are visible for communication.
- **Activity Logs:**
  Every submission, edit, or approval is logged to maintain transparency and accountability [11].

- **Research Link:**
  Security and accountability measures align with best practices in educational systems that involve multi-user evaluations [4][11].

User Interface (UI) Requirements

The platform's user interface is designed with simplicity and inclusivity in mind.

- **Responsiveness:**
  The design adapts across desktops, tablets, and smartphones using Tailwind's flexible layout utilities.

- **Ease of Navigation:**
  Key modules *Home, Books, Submit, Reviews, Profile, and Contact* are clearly visible and easy to access.

- **Accessibility:**
  Following accessibility standards, the platform uses semantic HTML, proper color contrast, and keyboard navigation to support all users.
- **Design Philosophy:**
  The UI presents information using card-based layouts, highlighting essential details like title, author, and average rating. Review forms are concise but informative, making evaluation faster and easier.
- **User Engagement:**
  Each section is designed to minimize effort and maximize clarity, in line with studies emphasizing usability and learner engagement [2][6][9].

4.2 Block diagram

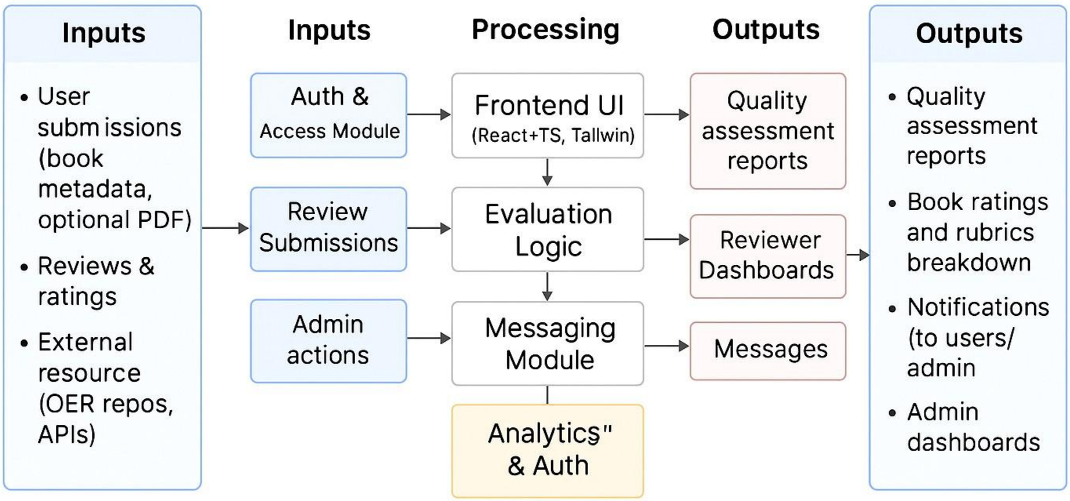

Inputs
• User subm issions (book metadata, optional PDF)
• Reviews & ratings
• External resource (OER repos, APIs)
Inputs
Processing
Outputs
Auth & Access Module
Frontend UI (React+TS, Tallwin)
Quality assessment reports
Review Submissions
Evaluation Logic
Reviewer Dashboards
Admin actions
Messaging Module
Messages
Analytics" & Auth
Outputs
• Quality assessment reports
• Book ratings and rubrics breakdown
• Notifications (to users/ admin
• Admin dashboards

Fig 4.1 Functional block diagram

Inputs

- **User submissions:** These include textbook details such as title, author, edition, subject, and optional digital copies (PDFs). Users like students and professors upload or submit books to be evaluated.

- **Review submissions:** Registered users contribute ratings, comments, and structured rubric-based evaluations that measure a textbook's clarity, curriculum alignment, and usefulness.
- **Admin actions:** Administrators monitor the overall system, approve or reject reviews, and manage user roles.
- **External resources:** Optional data from Open Educational Resource (OER) repositories or academic APIs can be integrated to enrich textbook metadata and context.

Processing Units

1. **Authentication & Access Module:**
   Managed by **Supabase Auth**, this module ensures that only authorized users can log in or make changes. It uses **role-based access control** to distinguish between students, professors, reviewers, and admins.

2. **Frontend UI (React + TypeScript + Tailwind):**
   The user interface is built using React and Tailwind CSS, offering a modern, responsive experience. It allows users to navigate through modules such as book submission, reviews, messages, and dashboards with ease.
3. **Evaluation Logic:**
   This part applies the evaluation rubrics and quality criteria—such as readability, alignment with curriculum, inclusivity, and presentation—to calculate a standardized textbook quality score. It ensures that every review follows consistent evaluation rules.
4. **Messaging Module:**
   The messaging feature supports communication among users. Students can discuss with professors or reviewers directly through real-time channels powered by **Supabase Realtime DB hooks**. This encourages collaboration and clarification during the review process.
5. **Analytics & Auth Module:**
   This final processing block gathers all evaluations and review data to generate visual insights. It creates analytical dashboards, charts, and summary reports that help users and admins identify top-rated textbooks and understand overall trends.

Outputs

- **Quality Assessment Reports:**
  These reports summarize each book’s overall score and detailed rubric-based evaluation.

- **Reviewer Dashboards:**
  Reviewers and professors can access dashboards showing their submissions, review statistics, and book comparisons.

- **Messages:**
  Notifications and message exchanges keep users updated about approvals, feedback, and ongoing discussions.
- **Admin Dashboards:**
  Administrators have access to control panels showing total users, reviews, pending approvals, and system analytics for effective management.

Suitability for the Project

- It clearly separates **input**, **processing**, and **output** stages, reflecting the real architecture used in the React + Supabase implementation.
- The modular design makes the system **scalable**, allowing new features such as AI-based evaluation or automatic plagiarism detection to be added easily in the future.
- Built-in **security and authentication** ensure that data remains protected while maintaining transparency.
- The inclusion of **messaging and analytics** modules demonstrates the system's focus on collaboration and evidence-based decision-making.

Suitability for the project:

The block diagram is suitable for this project as it clearly separates the flow of information from inputs to outputs through logical processing units. It ensures transparency in how data is collected, analyzed, and transformed into actionable insights. The modular structure allows for scalability (adding new criteria or evaluation methods in future), ensures security of user contributions, and provides a structured way to integrate both automated analysis and human feedback. This makes the block diagram an effective representation of the *Textbook Quality Assessment System* workflow.

4.3 System Flow chart

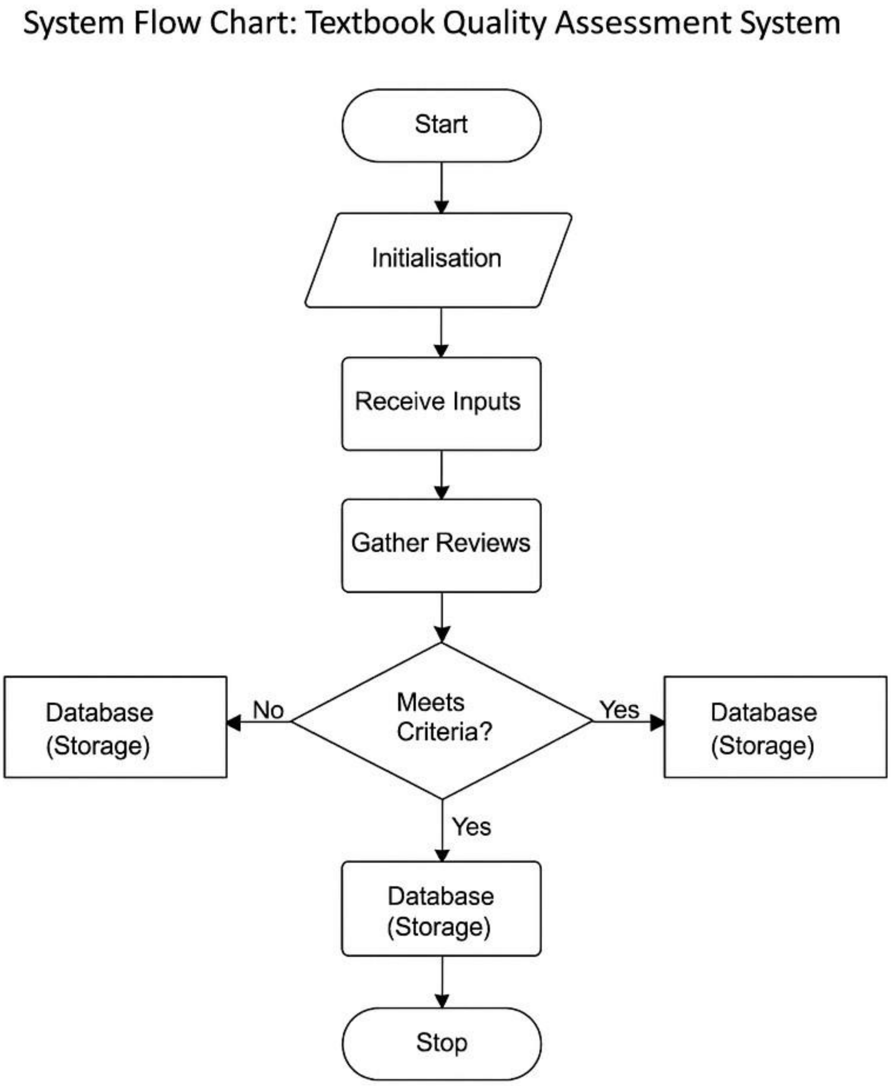


Fig 4.2 System flow chart

The flow begins with the **initialization phase**, where the system establishes its working environment. During this stage, the platform loads necessary frontend components (React modules), initializes the backend connection with **Supabase**, and verifies user authentication. This ensures that all system dependencies and resources are ready before accepting any user interactions.

Once initialization is complete, the **input phase** begins. Here, different users—students, professors, reviewers, and administrators submit data through the web interface. These inputs include textbook details, structured reviews, and comments based on predefined evaluation rubrics. The system validates each input to ensure completeness and correctness before moving forward.

After data submission, the system proceeds to the **processing phase**. In this stage, the application performs data validation, checks for duplicate or invalid records, and transfers approved inputs to the **evaluation logic module**. This module applies standardized rubrics such as *clarity, curriculum alignment, inclusivity,* and *presentation quality* to calculate the textbook's overall score. The processing logic ensures that every evaluation is consistent, fair, and traceable.

If the submitted data meets the evaluation criteria, it is then stored securely in the **Supabase PostgreSQL database**. This includes all book metadata, review records, and associated analytics. If a submission fails validation (for example, missing details or invalid file formats), the system prompts users to correct and resubmit.

The final stage is the **output generation** phase. Here, the system compiles the processed data and generates various results, including visual dashboards, summary analytics, and notification alerts. These outputs are displayed to different user roles according to their permissions: students view their contributions, professors access class-related reviews, and administrators monitor overall platform activity.

By the end of the flow, the system ensures that all valid information is stored, analyzed, and presented effectively to enhance transparency and improve decision-making about textbook selection and quality assurance.

Suitability for the Project

This flow chart is well-suited to represent the working model of the **TextAssess system** because it clearly demonstrates the logical sequence of operations from data input to final output.
It also reflects how each phase connects seamlessly in the **React + Supabase full-stack environment**, maintaining data accuracy, real-time synchronization, and security.

The structured flow promotes clarity in understanding system behavior and provides a strong foundation for scalability making it easier to add features like AI-assisted evaluation, plagiarism detection, or multi-language support in the future.

4.4 Choosing Devices

Although TextAssess is primarily a web-based platform and does not depend on specialized hardware, selecting appropriate devices for development and usage is still important. The system is designed to be accessible from any modern device—laptops, desktops, tablets, and smartphones. During development, a laptop with sufficient processing power, memory, and internet connectivity was chosen to ensure smooth execution of React components and Supabase interactions. On the user side, the platform works seamlessly on typical student and faculty devices, making accessibility universal. This flexibility reduces dependency on specific hardware and ensures wide adoption across academic environments.

4.5 Designing Units

- **Authentication Unit:** Handles signup, login, session management, and user role validation using Supabase Auth.
- **Book Submission Unit:** Allows users to upload textbook metadata and optional files.
- **Review Management Unit:** Collects reviews, applies rubrics, and stores scores.
- **Messaging Unit:** Enables communication between users regarding textbook clarifications.
- **Admin Unit:** Manages approvals, dashboards, and system-level decisions.
- **Storage Unit:** Manages file uploads and links through Supabase Storage.

4.6 Standards

- **UI/UX Standards:** Consistent layout, responsive design, semantic HTML, WCAG accessibility, and Tailwind design patterns.
- **Coding Standards:** TypeScript strict mode, modular React components, error handling, and naming conventions.
- **Security Standards:** Enforced Supabase Row-Level Security (RLS), encrypted authentication, strong password policies, and access restrictions.
- **Database Standards:** Normalized schema, foreign key relationships, indexing for performance, and JSONB usage for rubrics.
- **Quality Standards:** Structured reviews aligned with textbook evaluation research, ensuring academic relevance.

4.7 Mapping with IoTWF Reference Model Layers

- **Edge Layer:** User devices (laptops/phones) act as the interface points.
- **Communication Layer:** HTTPS requests sent through Supabase SDK.

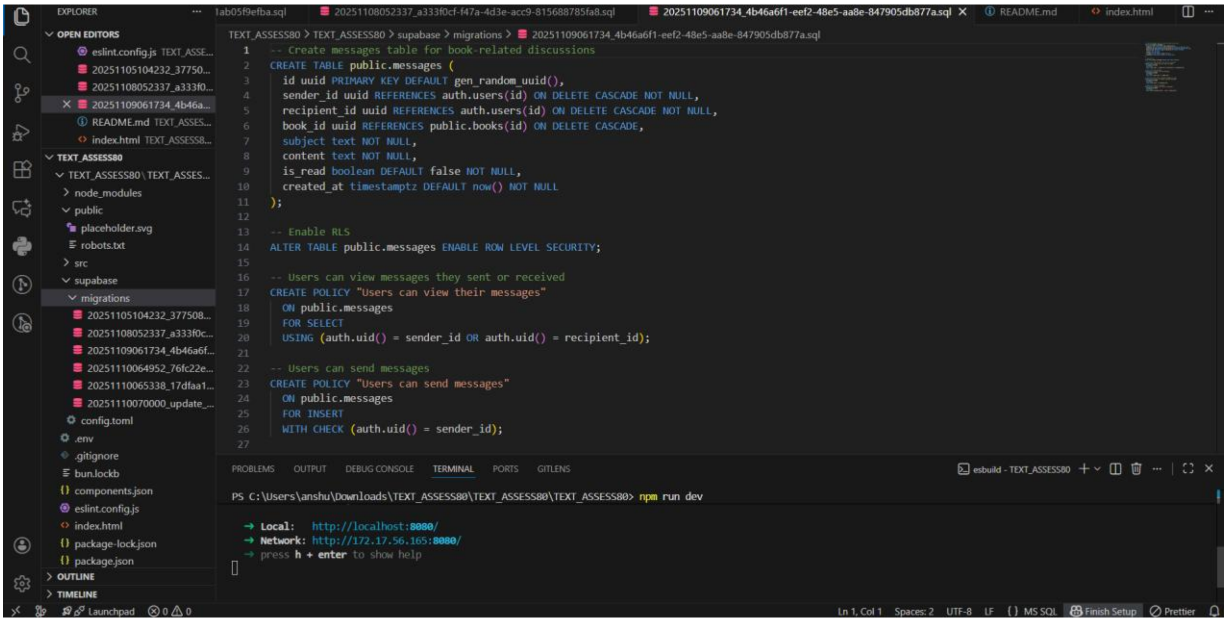
- **Data Layer:** Supabase PostgreSQL stores books, reviews, and messages.
- **Control Layer:** RLS policies and role-based access manage system control.
- **Application Layer:** React frontend delivering interactive features.
- **Collaboration Layer:** Messaging, reviews, and admin workflows supporting academic collaboration.

4.8 Domain Model Specification

- **User:** Represents students, reviewers, professors, and admins with specific roles.
- **Book:** Contains metadata such as title, author, edition, description, and file reference.
- **Review:** Stores scores, rubrics, comments, and reviewer identity.
- **Message:** Exchanges communication between users.
- **Admin:** Approves reviews, manages activities, and oversees system consistency.

Relationships include:

- A user can submit multiple books.
- A book can have multiple reviews.
- A review belongs to one book and one reviewer.
- Messages occur between two users.



```sql
-- Create messages table for book-related discussions
CREATE TABLE public.messages (
  id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
  sender_id uuid REFERENCES auth.users(id) ON DELETE CASCADE NOT NULL,
  recipient_id uuid REFERENCES auth.users(id) ON DELETE CASCADE NOT NULL,
  book_id uuid REFERENCES public.books(id) ON DELETE CASCADE,
  subject text NOT NULL,
  content text NOT NULL,
  is_read boolean DEFAULT false NOT NULL,
  created_at timestamptz DEFAULT now() NOT NULL
);

-- Enable RLS
ALTER TABLE public.messages ENABLE ROW LEVEL SECURITY;

-- Users can view messages they sent or received
CREATE POLICY "Users can view their messages"
  ON public.messages
  FOR SELECT
  USING (auth.uid() = sender_id OR auth.uid() = recipient_id);

-- Users can send messages
CREATE POLICY "Users can send messages"
  ON public.messages
  FOR INSERT
  WITH CHECK (auth.uid() = sender_id);
```

```
PS C:\Users\anshu\Downloads\TEXT_ASSESS80\TEXT_ASSESS80\TEXT_ASSESS80> npm run dev

  ➜  Local:   http://localhost:8080/
  ➜  Network: http://172.17.56.165:8080/
  ➜  press h + enter to show help
```

Fig 4.3 Database Schema for Books and Reviews in TextAssess

4.9 Communication Model

1. **Frontend → Backend Communication:**

   ○ Through Supabase client for CRUD operations.
   ○ Uploading files, submitting reviews, logging in, and sending messages.

2. **Backend → Frontend Communication:**

   ○ Realtime updates when new reviews or messages arrive.
   ○ Authentication status changes (session events).
   ○ Data retrieval on demand via database queries.

3. **Internal Communication:**

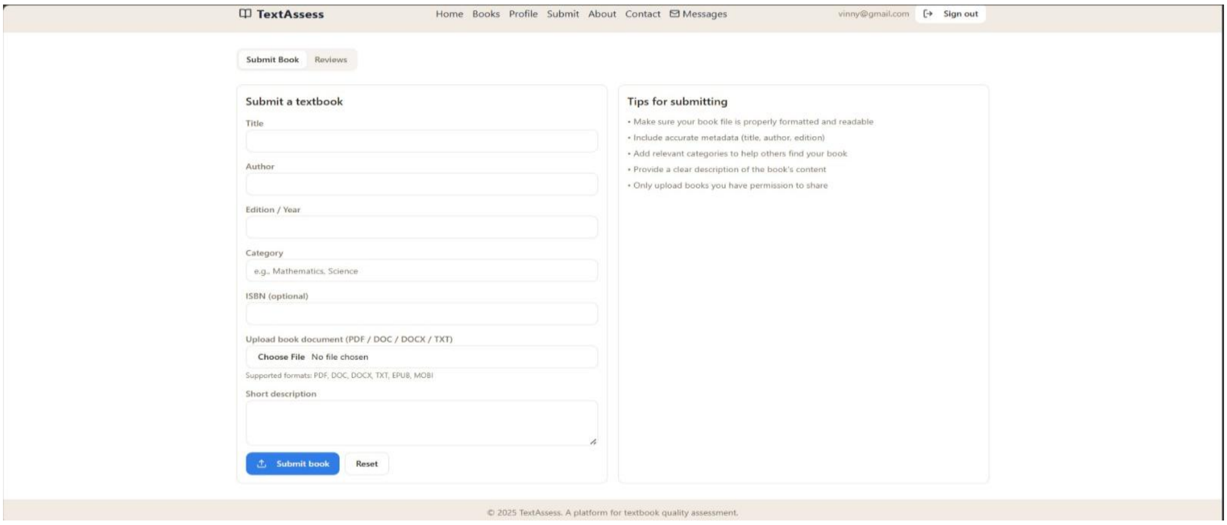
   ○ Supabase Storage interacts with DB using file paths.
   ○ Policies regulate which roles can access which data.

4.10 IoT Deployment Level:

- **Level 1: Device Interaction** – Users interact through personal devices.
- **Level 2: Data Transmission** – Data is sent securely through Supabase SDK.
- **Level 3: Data Storage** – All system data is stored in Supabase PostgreSQL.
- **Level 4: Data Processing** – Reviews, analytics, role validation, and messages are processed.
- **Level 5: System Application** – Frontend displays dashboards, evaluations, and results.
- **Level 6: Collaboration** – Messaging and shared evaluations support academic collaboration.

4.11 Functional View

- **User Authentication:** Verifies identity and ensures secure login.
- **Book Submission:** Allows uploading metadata and files.
- **Review Collection:** Accepts structured evaluations and rubrics.
- **Messaging:** Facilitates communication between users.
- **Admin Control:** Approves content, monitors system, and manages users.
- **Analytics:** Computes averages, scores, and provides insights.
- **File Management:** Handles uploading, storing, and accessing PDFs.



Submit a textbook

Title

Author

Edition / Year

Category

e.g., Mathematics, Science

ISBN (optional)

Upload book document (PDF / DOC / DOCX / TXT)

Choose File No file chosen

Supported formats: PDF, DOC, DOCX, TXT, EPUB, MOBI

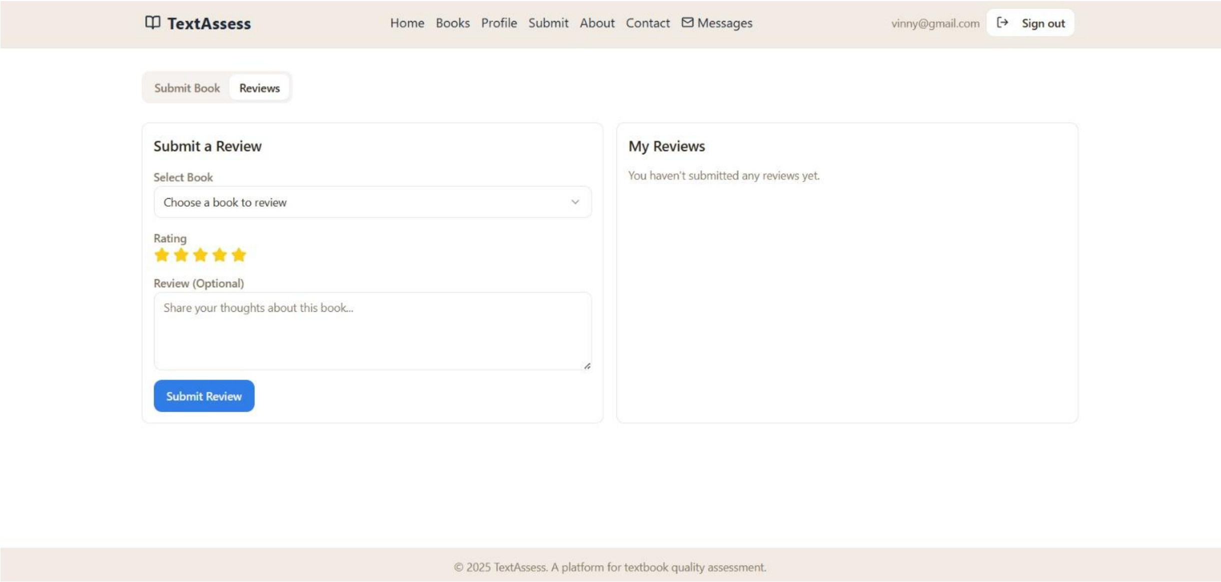
Short description

Tips for submitting

- Make sure your book file is properly formatted and readable
- Include accurate metadata (title, author, edition)
- Add relevant categories to help others find your book
- Provide a clear description of the book's content
- Only upload books you have permission to share

© 2025 TextAssess. A platform for textbook quality assessment.

Fig 4.4 Book Submission Module



Submit a Review

Select Book

Choose a book to review

Rating

★★★★★

Review (Optional)

Share your thoughts about this book...

My Reviews

You haven't submitted any reviews yet.

© 2025 TextAssess. A platform for textbook quality assessment.

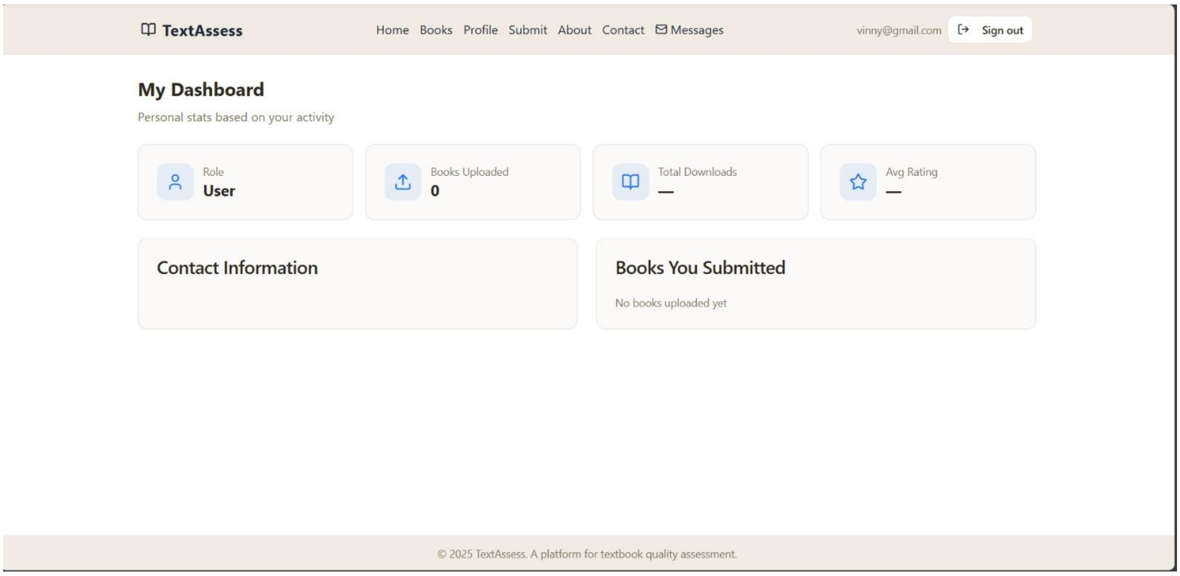
Fig 4.5 Review Submission-Rating and Comment

4.12 Mapping IoT Deployment Level with Functional View

- **Device Interaction → Authentication & User Input**
- **Data Transmission → Communication with Supabase**
- **Data Storage → Books, Reviews, Users, Messages tables**
- **Data Processing → Analytics and Review Rubrics**
- **Application Layer → React UI for students, reviewers, and admin ● Collaboration Layer → Messaging and multi-role interactions**

4.13 Operational View

- Users log in through the web interface using Supabase Auth.
- Students and professors submit books or reviews, which are validated and stored in the database.
- Reviewers evaluate textbooks based on rubrics synthesized from academic research.
- Admins approve or reject submissions and monitor overall system activity.
- Realtime functionality ensures immediate updating of dashboards and messages.
- The system maintains security through authentication, RLS policies, and controlled access.
- The platform operates continuously with minimal manual intervention due to its cloud-based infrastructure.



My Dashboard

Personal stats based on your activity

Role: User

Books Uploaded: 0

Total Downloads: —

Avg Rating: —

Contact Information

Books You Submitted

No books uploaded yet

© 2025 TextAssess. A platform for textbook quality assessment.

Fig 4.6 User Dashboard – Profile and Stats

Chapter 5

HARDWARE,SOFTWARE AND SIMULATION

5.1 Hardware

The TextAssess project is a cloud-based web application and therefore does not depend on specialized hardware components. However, reliable hardware was still necessary during the development and testing phases. The system was developed using a standard personal laptop with moderate specifications, such as an Intel or AMD processor, 8 GB RAM, and stable internet connectivity. This ensured that the React development environment, Node.js, and Supabase dashboard ran smoothly without performance issues.

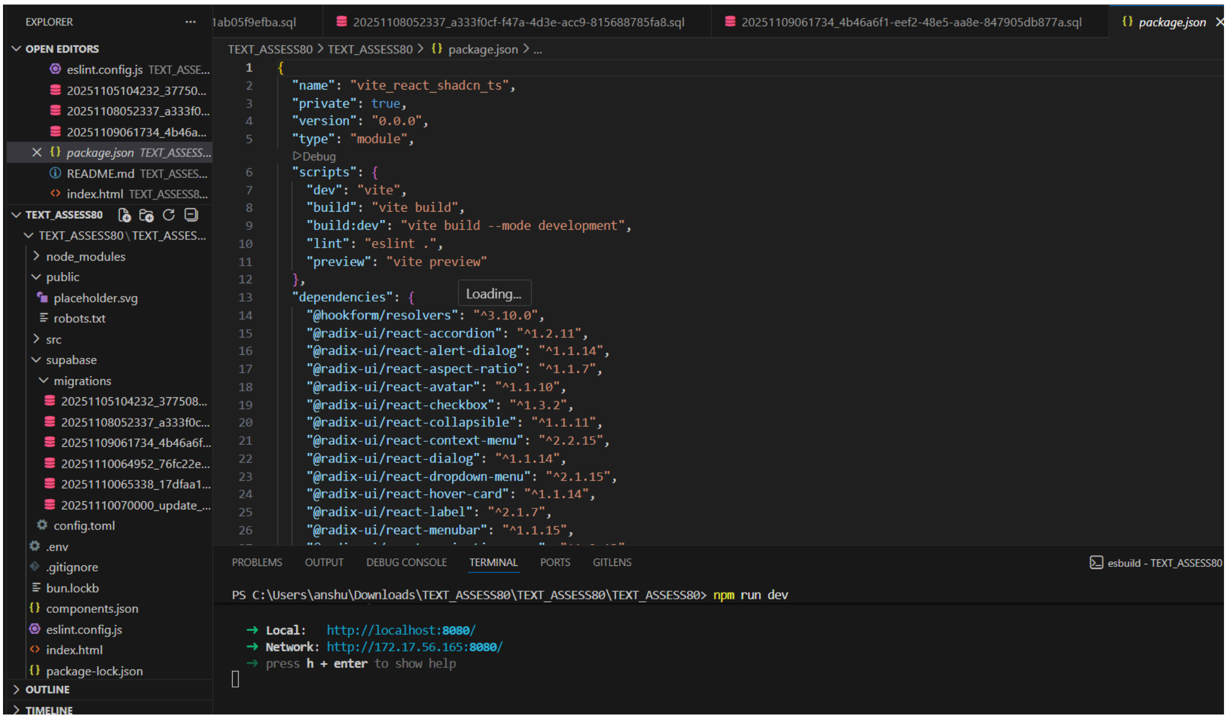
For end-users including students, professors, reviewers, and administrators the only requirement is a basic device that can run a modern web browser. The platform works seamlessly on laptops, desktops, tablets, and smartphones. This hardware independence makes the system highly accessible and reduces the barriers to adoption across educational institutions.

5.2 Software Development Tools

The project relies entirely on open-source and cloud-based software tools, making it efficient, cost-effective, and easy to maintain.

The frontend was developed using **React.js** and **TypeScript**, which ensured component reusability, type safety, and predictable behavior. **Tailwind CSS** and **shadcn/UI** were used to design a clean, responsive, and accessible user interface. Navigation and routing were handled using **React Router**, while **React Query** was used to manage data fetching and caching. On the backend, **Supabase** served as the primary platform, offering authentication, database management (PostgreSQL), file storage, and realtime updates. Additional tools such as **Node.js**, **npm**, **VS Code**, **Git**, and **GitHub** supported development, version control, and collaboration.

These software tools collectively ensured smooth development, fast debugging, and seamless integration between the frontend and backend.



```json
{
  "name": "vite_react_shadcn_ts",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "build:dev": "vite build --mode development",
    "lint": "eslint .",
    "preview": "vite preview"
  },
  "dependencies": {
    "@hookform/resolvers": "^3.10.0",
    "@radix-ui/react-accordion": "^1.2.11",
    "@radix-ui/react-alert-dialog": "^1.1.14",
    "@radix-ui/react-aspect-ratio": "^1.1.7",
    "@radix-ui/react-avatar": "^1.1.10",
    "@radix-ui/react-checkbox": "^1.3.2",
    "@radix-ui/react-collapsible": "^1.1.11",
    "@radix-ui/react-context-menu": "^2.2.15",
    "@radix-ui/react-dialog": "^1.1.14",
    "@radix-ui/react-dropdown-menu": "^2.1.15",
    "@radix-ui/react-hover-card": "^1.1.14",
    "@radix-ui/react-label": "^2.1.7",
    "@radix-ui/react-menubar": "^1.1.15",
```

```
PS C:\Users\anshu\Downloads\TEXT_ASSESS80\TEXT_ASSESS80\TEXT_ASSESS80> npm run dev

  ➜  Local:   http://localhost:8080/
  ➜  Network: http://172.17.56.165:8080/
  ➜  press h + enter to show help
```

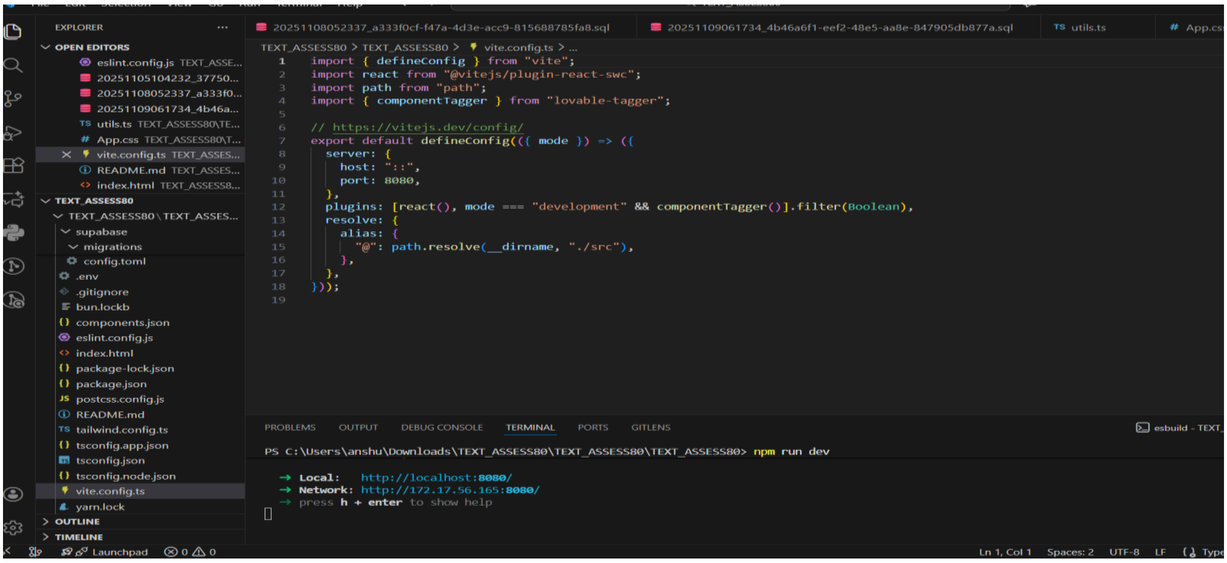
Fig 5.1 Frontend Dependency Configuration

5.3 Software Code

The software implementation consists of modular React components connected to the Supabase backend through the official supabase-js client. Each feature, such as authentication, book submission, review management, analytics, and admin approval, was developed as an independent module to ensure clarity and maintainability.

The code structure follows best practices, including reusable hooks, separate service files for database queries, validation logic for forms, and asynchronous calls for data operations. Supabase Auth manages user sessions, while database CRUD operations handle book updates, review insertions, and message exchanges.

Although the complete codebase contains numerous components, the core logic emphasizes readability, minimal redundancy, and strict TypeScript checks. This ensures that the application performs consistently and can be extended easily in the future.



```
import { defineConfig } from "vite";
import react from "@vitejs/plugin-react-swc";
import path from "path";
import { componentTagger } from "lovable-tagger";

// https://vitejs.dev/config/
export default defineConfig(({ mode }) => ({
  server: {
    host: "::",
    port: 8080,
  },
  plugins: [react(), mode === "development" && componentTagger()].filter(Boolean),
  resolve: {
    alias: {
      "@": path.resolve(__dirname, "./src"),
    },
  },
}));
```

```
PS C:\Users\anshu\Downloads\TEXT_ASSESS80\TEXT_ASSESS80\TEXT_ASSESS80> npm run dev

  → Local:   http://localhost:8080/
  → Network: http://172.17.56.165:8080/
  → press h + enter to show help
```

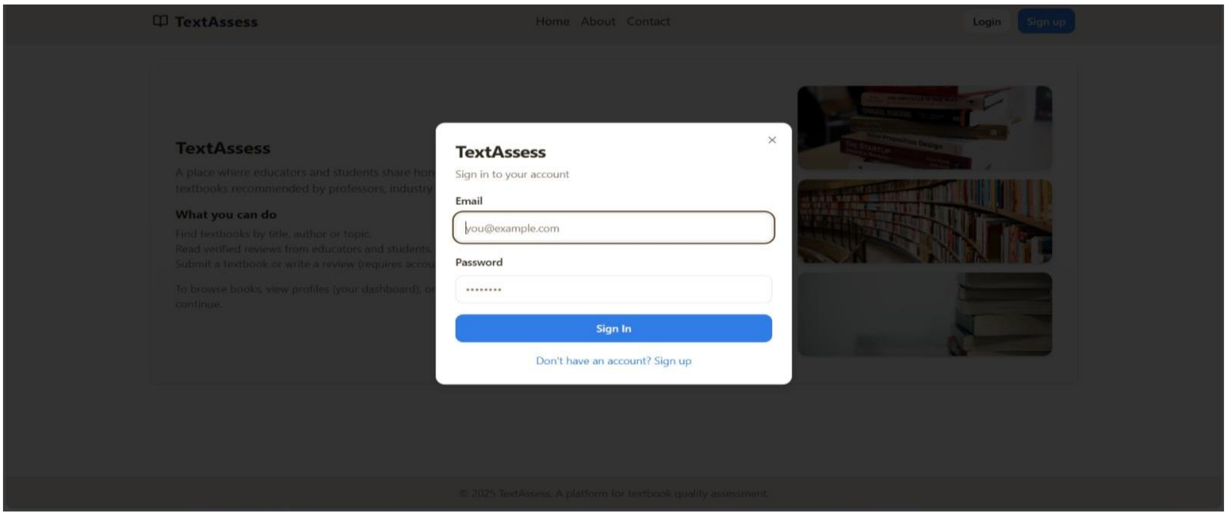
Fig 5.2 Vite Build and Development Configuration

5.4 Simulation

Since TextAssess is a web application rather than an embedded or IoT-based system, simulation refers to evaluating system functionality in a controlled test environment. The simulation environment consisted of running the frontend locally on a development server while connecting to a live Supabase backend. This setup allowed developers to simulate real user interactions such as logging in, submitting reviews, uploading book files, sending messages, and navigating through dashboards.

Test data representing sample books, dummy accounts, and mock reviews was inserted to verify how the system behaves under different scenarios. Realtime features like messaging and review updates were also simulated to ensure that the platform responded instantly to changes.

This simulated environment replicated real-world usage, allowing early detection of bugs, performance bottlenecks, and UI inconsistencies before deployment.



TextAssess

A place where educators and students share hon
textbooks recommended by professors, industry

What you can do

Find textbooks by title, author or topic.
Read verified reviews from educators and students.
Submit a textbook or write a review (requires accou

To browse books, view profiles (your dashboard), or
continue.

TextAssess

Sign in to your account

Email

you@example.com

Password

••••••••

© 2025 TextAssess. A platform for textbook quality assessment.

Fig 5.3 User Sign in Interface

Chapter 6

EVALUATION AND RESULTS

6.1 Test Points

- User authentication and role-based access
- Book submission and file upload
- Review creation, validation, and storage
- Admin approval workflows
- Messaging and realtime updates
- Database operations and data consistency
- User interface responsiveness and navigation
- Security checks using Supabase Row-Level Security (RLS)

6.2 Test Plan

The project followed a structured test plan aligned with the V-Model methodology. Testing began at the unit level, where individual React components and Supabase queries were verified for correct behavior. This was followed by integration testing to check if multiple modules such as review forms and dashboards worked cohesively.

System testing validated complete workflows, such as a student logging in, submitting a book, writing a review, and the admin approving it. Acceptance testing was carried out by simulating actual academic use cases to verify whether the system met the needs of students, professors, and reviewers.

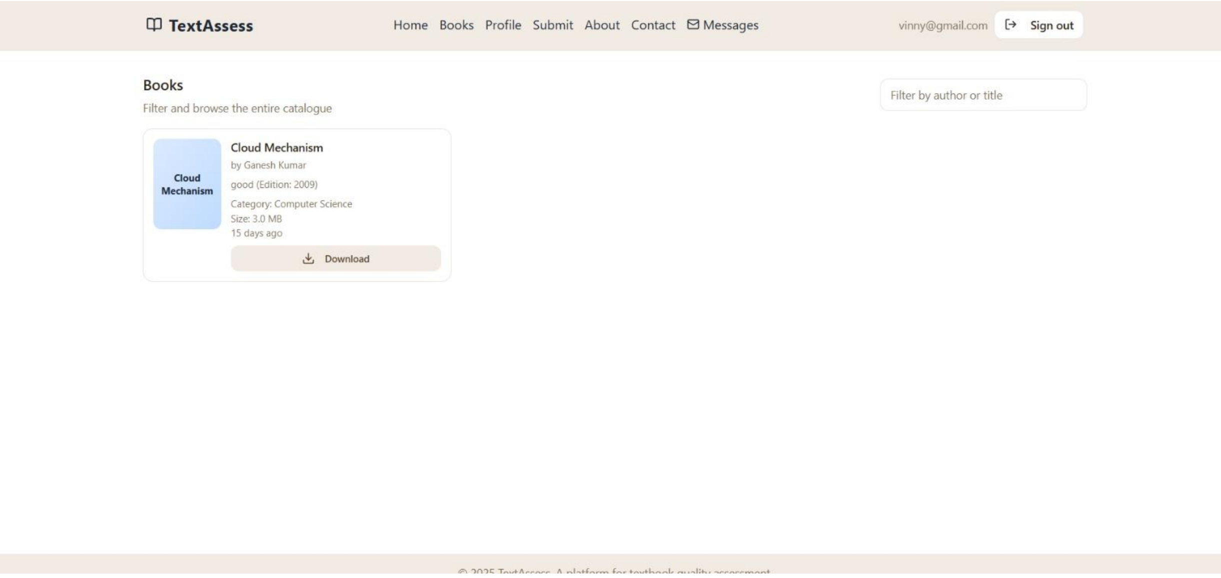
Each test scenario included input validation, error handling, edge cases, and user behavior testing to ensure overall stability.

6.3 Test Results

Users were able to register, log in, and access role-specific dashboards without issues. Book uploads and review submissions worked smoothly, with proper validation and secure data storage. Realtime features such as messaging and review updates responded instantly, demonstrating the strength of the Supabase backend.

The admin panel successfully displayed pending reviews, book submissions, and analytical summaries. All approved reviews were accurately reflected in the textbook rating system, and the analytics module generated consistent average scores across subjects.

Minor UI inconsistencies and validation improvements were identified during testing but were fixed immediately. Overall, the system achieved all functional requirements and operated as expected.



Books

Filter and browse the entire catalogue

Filter by author or title

Cloud Mechanism

Cloud Mechanism

by Ganesh Kumar

good (Edition: 2009)

Category: Computer Science

Size: 3.0 MB

15 days ago

© 2025 TextAssess. A platform for textbook quality assessment

Fig 6.1 Successful Rendering of Books Catalogue

6.4 Insights

- The use of Supabase significantly reduced backend complexity while still offering strong authentication and database security.
- Real-time features greatly improved collaboration, making the platform feel dynamic and interactive.
- Structured rubrics provided meaningful and consistent evaluation metrics across different reviewers.
- The modular React architecture made debugging and enhancement extremely efficient.
- The system is highly scalable and can support multiple institutions with minimal modifications.
- The platform successfully brings transparency to textbook evaluation, aligning with research findings from the referenced papers.

Chapter 7

SOCIAL,LEGAL,ETHICAL,SUSTAINABILITY AND SAFETY ASPECTS

7.1 Social Aspects

TextAssess has a significant social influence because it transforms how educational institutions evaluate and adopt textbooks. Traditionally, the process of selecting textbooks happens behind closed doors, usually by a small committee. Students the primary users of textbooks rarely have the opportunity to express their opinions, resulting in materials that may not always meet their learning needs.

TextAssess directly addresses this gap by giving every stakeholder a voice. By allowing students, professors, and reviewers to contribute evaluations in an open and transparent manner, the platform encourages a more democratic and inclusive decision-making process. This helps institutions better understand what types of textbooks create effective learning experiences, especially for diverse groups of students.

The platform also promotes social equality. Students from different academic backgrounds, social environments, or skill levels can participate equally because the system is accessible online and does not require advanced hardware or technical skills. A student with only a mobile phone has the same ability to provide feedback as someone with a high-end laptop.

Additionally, TextAssess enhances communication between students and educators. When professors review feedback submitted by their students, they gain valuable insights into the challenges faced in understanding certain textbooks. This fosters healthy academic dialogue and supports a more student-centered learning environment.

Thus, TextAssess not only improves textbook quality but also strengthens the social fabric of educational institutions by promoting fairness, openness, and meaningful collaboration.

7.2 Legal Aspects

Since TextAssess handles user information, uploaded files, and academic reviews, it must operate responsibly within applicable legal frameworks. The platform makes use of Supabase Auth, which follows modern security standards to protect login credentials through encrypted storage and secured authentication flows. This helps the system comply with basic data protection regulations and prevents unauthorized individuals from accessing personal user information.

Another important legal factor is intellectual property. Users may upload book samples, notes, or reference materials during the review process. The system encourages users to upload only content they have permission to share, protecting the rights of authors and publishers. This reduces the risk of copyright infringement and ensures that the platform remains legally sound.

The system also avoids collecting unnecessary data. Only essential fields such as name, email, and role are stored, helping the platform adhere to the principle of data minimization. Clear usage guidelines are provided to users, instructing them to avoid offensive, defamatory, or harmful content.
By ensuring secure authentication, respecting copyright limitations, and promoting responsible user behavior, TextAssess remains compliant with legal expectations and performs as a safe academic tool.

7.3 Ethical Aspects

Ethical responsibility is one of the most important aspects of the TextAssess platform. Because the system deals with reviews and ratings, it must ensure fairness and prevent biased or harmful feedback. To achieve this, the platform uses structured rubrics that guide reviewers to evaluate textbooks based on academic criteria such as clarity, difficulty level, accuracy, accessibility, and relevance rather than personal judgments.

The system also protects user privacy by ensuring that sensitive information is visible only to necessary personnel, such as admins. Users are encouraged to provide genuine and constructive feedback. Any form of discrimination whether based on gender, race, academic ability, or background is strictly discouraged. Ethical guidelines are communicated clearly so that all users understand the importance of respectful and professional behavior.

Another ethical challenge in online review systems is the possibility of fake or misleading reviews. TextAssess reduces this risk by verifying user identities through Supabase Auth and assigning roles such as student, reviewer, professor, or admin. This ensures that only genuine users contribute evaluations and that administrators can moderate inappropriate content.

By encouraging honesty, fairness, transparency, and accountability, TextAssess promotes an ethical learning environment where feedback is used to improve education and empower learners.

7.4 Sustainability Aspects

TextAssess contributes positively to sustainability on several levels—environmental, operational, and educational.
Environmentally, the system eliminates the need for paper-based review forms, physical reports, and manually maintained documentation. Textbook evaluation committees often rely heavily on printouts and paper surveys. Moving this process online reduces paper consumption and aligns with global efforts toward eco-friendly digital transformation.

The system also benefits from the cloud-based infrastructure of Supabase. Instead of requiring physical servers or hardware maintenance, TextAssess runs on efficiently managed, optimized cloud servers shared by multiple applications. This reduces energy consumption and carbon emissions when compared to dedicated physical hardware.

Operational sustainability is achieved because the platform is easy to update, scale, and maintain. When institutions grow or new subjects are introduced, the system can accommodate new users or additional data without requiring major upgrades. This long-term maintainability ensures that the platform remains effective for future academic needs.

Finally, TextAssess promotes educational sustainability by helping institutions continuously select high-quality textbooks. Poor-quality or outdated materials can negatively impact learning outcomes for years. By using structured evaluations, the platform ensures that only the most effective textbooks are chosen, supporting sustainable academic growth and long-term student success.

7.5 Safety Aspects

Safety within TextAssess focuses on secure usage, preventing data misuse, and protecting users from harmful or inappropriate content. Supabase provides strong authentication and database security mechanisms that create a safe and reliable foundation for the platform. Features such as encrypted passwords, protected user sessions, and identity verification ensure that only legitimate users can access their own accounts.

Row Level Security in Supabase adds an important layer of protection by controlling exactly who can view or modify data. This means a student cannot see another student’s private messages, reviews, or personal information. Administrators have higher permissions, but even they operate within clearly defined safety boundaries.

The system also incorporates content moderation. Since users are allowed to upload files and submit written feedback, all sensitive content passes through an admin approval process. This ensures that no harmful, abusive, misleading, or inappropriate material becomes visible to other users. This is especially important when the system is accessed by learners of different age groups.

On the technical side, the platform safeguards data against accidental loss or corruption. Supabase manages file storage and database backups, ensuring that user information and uploaded documents remain secure even in the event of unexpected system issues.

Overall, TextAssess offers a safe digital environment where students, educators, and reviewers can confidently interact with the system without worrying about security threats, data misuse, or exposure to harmful content.

Chapter 8

CONCLUSION

The project successfully developed and implemented **TextAssess**, a comprehensive **Textbook Quality Assessment System** designed to streamline the evaluation of academic textbooks through structured criteria, collaborative reviews, and automated analysis.
The system integrates **feedback from students, professors, and reviewers**, ensuring that evaluations reflect both pedagogical standards and learner perspectives, as recommended by previous research on textbook evaluation and quality frameworks [1][3][5][6][7].

Following the **V-Model methodology**, the project progressed through clear stages — from requirements gathering and literature review to system design, development, and testing. This structured approach ensured that each requirement was validated against outcomes, maintaining transparency and traceability throughout the development cycle [2][4][8].

The use of a **modern full-stack architecture (React + TypeScript frontend and Supabase backend)** provided a secure, scalable, and cloud-ready environment.

- The **frontend** focused on accessibility and user experience, allowing seamless interaction through structured forms, review dashboards, and feedback modules.
- The **backend**, powered by Supabase's PostgreSQL and real-time synchronization, handled authentication, secure data storage, and analytics, supporting multi-role collaboration between students, reviewers, and administrators.

By implementing **role-based access control**, **encryption policies**, and **real-time data validation**, the system ensured high standards of **data security and user privacy**, aligning with the best practices outlined in studies on educational technology systems [3][5][10].

The outcomes of this project confirm that all key objectives were successfully achieved:

- **Behavioral Objective:**
  Users across different roles (students, professors, reviewers, administrators) were able to interact effectively with the platform, submitting structured reviews and navigating dashboards intuitively. This encouraged consistent, research-backed evaluation behavior [1][6][9].

- **Analytical Objective:**
  The evaluation engine generated measurable indicators of textbook quality (clarity, curriculum relevance, inclusivity, and design) using standardized rubrics, as suggested in earlier frameworks by Mohammadi and Fey [3][4][7]. These results provided evidence-based insights for decision-makers in curriculum design and textbook selection.
- **System Management Objective:**
  The administrator dashboard enabled efficient review moderation, approval workflows, and reporting. This directly addresses Mahmood's [5] and Yasar's [8] findings on the importance of structured oversight in maintaining textbook evaluation integrity.
- **Security Objective:**
  Supabase authentication and database-level policies ensured safe, role-restricted access. User privacy was maintained by showing only verified email contacts, ensuring compliance with ethical standards for educational data management [5][10].
- **Deployment Objective:**

The prototype was successfully deployed in a local environment and designed for smooth migration to cloud infrastructure. Its scalability supports institutional-level implementation, consistent with Ivić's [7] emphasis on textbook evaluation systems that can be adapted across educational contexts.

REFERENCES

[1] G. W. Cobb, "Introductory textbooks: A framework for evaluation: A comparison of 16 books," *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 321–339, 1987.

[2] R. A. R. Gurung and R. C. Martin, "Predicting textbook reading: The textbook assessment and usage scale," *Teaching of Psychology*, vol. 38, no. 1, pp. 22–28, 2011.

[3] C.-C. Fey and E. Matthes, "Textbook quality criteria and evaluation," in *The Palgrave Handbook of Textbook Studies*, New York: Palgrave Macmillan US, 2018, pp. 157–167.

[4] M. Mohammadi and H. Abdi, "Textbook evaluation: A case study," *Procedia - Social and Behavioral Sciences*, vol. 98, pp. 1148–1155, 2014.

[5] K. Mahmood, "Conformity to quality characteristics of textbooks: The illusion of textbook evaluation in Pakistan," *Journal of Research and Reflections in Education*, vol. 5, no. 2, pp. 170–190, 2011.

[6] A. Woodward, D. L. Elliot, and K. C. Nagel, *Textbooks in School and Society: An Annotated Bibliography & Guide to Research*, New York: Routledge, 2013.

[7] I. Ivić, S. Antić, and A. Pešikan, eds., *Textbook Quality: A Guide to Textbook Standards*, vol. 2, Göttingen: V&R unipress GmbH, 2013.

[8] O. Yasar, "A comparative analysis of assessment and evaluation exercises included in geography textbooks written according to the 2005 secondary education geography curriculum and textbooks of the former curriculum in Turkey," *International Journal of Progressive Education*, vol. 5, no. 1, pp. 45–68, 2009.

[9] P. Race, *The Lecturer's Toolkit: A Practical Guide to Assessment, Learning and Teaching*, London: Routledge, 2019.

[10] J. N. Giger and L. Haddad, *Transcultural Nursing: Assessment and Intervention*, Elsevier Health Sciences, 2020.

Base Paper:

From References the mainly referred paper: G. W. Cobb, "Introductory textbooks: A framework for evaluation: A comparison of 16 books," *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 321–339, 1987.

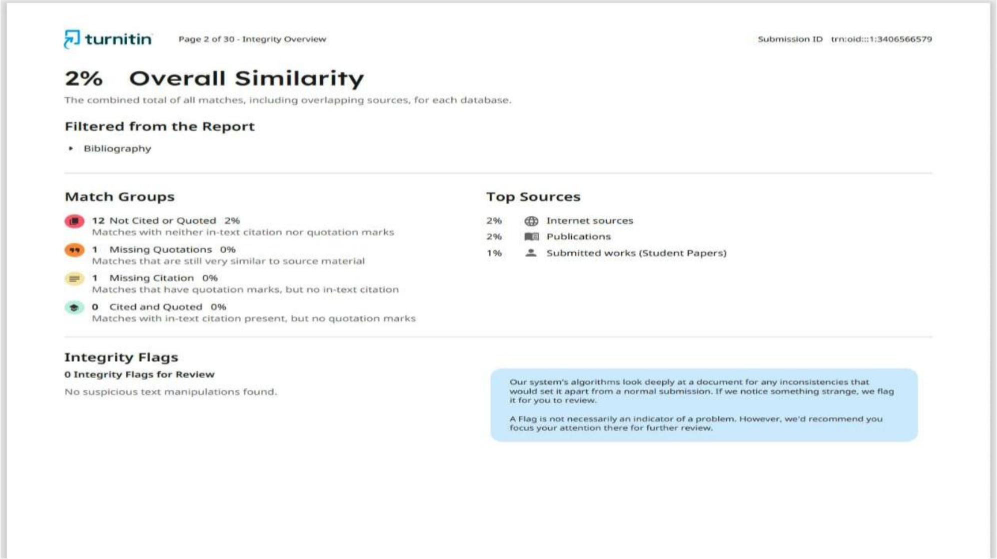
APPENDIX

i. Data Sheets

- Frontend Framework: React + TypeScript
- Build Tool: Vite
- Styling: Tailwind CSS
- Backend: Supabase (PostgreSQL + Storage + Auth)
- Authentication: JWT-based (Supabase Auth)
- Hosting (Local Dev): Vite at http://localhost:8080

ii. Project Report – Similarity Report

- Similarity Index: 2% (Turnitin range)



turnitin Page 2 of 30 - Integrity Overview Submission ID trn:oid:::1:3406566579

2% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

12 Not Cited or Quoted 2%
Matches with neither in-text citation nor quotation marks

1 Missing Quotations 0%
Matches that are still very similar to source material

1 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

2%	Internet sources
2%	Publications
1%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

a. Turnitin Similarity Report

iii. Live Project Demo

Git Hub link :

Live Demo :