

VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY  
THE INTERNATIONAL UNIVERSITY  
SCHOOL OF COMPUTER SCIENCE and ENGINEERING



**OnTest – An Online Testing System using React and Node.JS**

By

Ta Vi Khang – ITITIU20226

A thesis submitted to the School of Computer Science in partial fulfillment  
of the requirements for the degree of Bachelor of Computer Science and  
Engineering

Hochiminh city, Vietnam

2025

# **OnTest – An Online Testing System using React and Node.JS**

APPROVED BY:

MSc. Le Thanh Son

## **ACKNOWLEDGMENTS**

I am deeply grateful to my advisor, MSc. Le Thanh Son, for his unwavering support, invaluable guidance, and consistent patience throughout the course of this research endeavor. His flexible approach, as well as the additional time he so willingly provided to me was truly instrumental in reaching the conclusion of this project, and it also allowed me to learn at my own pace, while not compromising the scope or objectives of the thesis. His faith in me will never be forgotten.

This work also owes a special debt of gratitude to a dear friend, whose constant support, care, insightful advice and heartfelt encouragement has proven to be essential for getting to this point. The countless hours of help and guidance are gratefully acknowledged, as they greatly helped with completing this project, especially during some of the more challenging parts of the development.

Finally, I would like to express my deepest gratitude to my beloved family. Their unwavering support and constant encouragement helped me to overcome many challenges and to persevere through the more difficult periods of this research. I especially appreciate their understanding and patience during the many long nights and weekends that I dedicated to this project. Their love and support have been invaluable to me.

## TABLE OF CONTENTS

LIST OF TABLES .....	vi
LIST OF FIGURES .....	vii
ABSTRACT .....	viii
CHAPTER I INTRODUCTION.....	1
1.1 Background and Motivation .....	1
1.2 Problem Statement .....	2
1.3 Research Objectives.....	3
1.4 Scope of Research.....	4
1.5 Limitation.....	6
CHAPTER II LITERATURE REVIEW .....	7
2.1 Modern Educational Assessment Methods .....	7
2.1.1 Traditional vs Online Assessment .....	7
2.1.2 Effective Purpose of Various Question Types.....	8
2.2 Existing Online Assessment Systems .....	8
2.2.1 Analysis of Popular Platforms .....	8
2.2.2 Market Gaps and Opportunities .....	9
2.3 Automated Grading Algorithms .....	10
2.3.1 Automated Grading Accuracy .....	10
2.3.2 OnTest's Method for Improving Accuracy.....	10
2.3.3 Methods of Grading Various Question Types .....	10
2.4 Learning Analytics.....	11
2.4.1 Key Metrics for Evaluating Learning Effectiveness.....	11
2.4.2 Using Data to Improve Learning Outcomes .....	11
CHAPTER III METHODOLOGY .....	13
3.1 Development Process.....	13
3.2 System Requirements Analysis.....	14
3.2.1 Functional Requirements .....	15
3.2.2. Non-functional Requirements.....	26
3.3 System and Database Design .....	27

3.3.1 System Architecture.....	28
3.3.2 Database Design.....	30
CHAPTER IV IMPLEMENT.....	34
4.1 Assessment Creation and Management .....	34
4.1.1 Comprehensive Question Type Support .....	34
4.1.2 Quiz Creation Workflow.....	35
4.1.3 Class Management Integration .....	37
4.2 Test Participation and Submission Handling.....	38
4.2.1 Student Quiz Interface Design .....	38
4.2.2 Question Type Interaction Implementations.....	39
4.2.3 Submission Process and Integrity Assurance .....	39
4.3 Automated Grading and Manual Grading.....	40
4.3.1 Grading Algorithm Implementation .....	40
4.3.2 Manual Grading Interface for Essays .....	42
4.3.3 Score Calculation and Performance Analytics .....	43
CHAPTER V RESULTS .....	45
5.1 Results.....	45
5.2 Limitations and Future Enhancements.....	46
CHAPTER VI CONCLUSION .....	47
6.1 Conclusion .....	47
6.2 Recommendations.....	47
LIST OF REFERENCES.....	49

## LIST OF TABLES

Table	Page
1: Table 3.1: Iteration Breakdown and Major Activities	13
2: Table 3.2.1: Description of Use Cases in the OnTest System	16
3: Table 3.2.1.B1 – Authentication and Authorization Requirements	20
4: Table 3.2.1.B2 – Class Management Requirements	20
5: Table 3.2.1.B3 – Question Bank Management Requirements	21
6: Table 3.2.1.B4 – Quiz Management Requirements	21
7: Table 3.2.1.B5 – Assessment and Grading Requirements	22
8: Table 3.2.1.B6 – Reporting and Analytics Requirements	23
9: Table 3.2.1.C1 – Reporting and Analytics Requirements	24
10: Table 3.2.1.C2 – Class Engagement Requirements	24
11: Table 3.2.1.C3 – Quiz Taking Requirements	24
12: Table 3.2.1.C4 – Reporting and Analytics Requirements	25
13: Table 3.2.2 – Non-functional Requirements	26
14: Table 3.3.2: Analysis of Key Entities and Their Relationships	32

## LIST OF FIGURES

Figures	Page
1: <b>Figure 3.1:</b> Iterative Development Cycle	14
2: <b>Figure 3.2.1:</b> Use Case Diagram of the OnTest System	16
3: <b>Figure 3.3.1:</b> 3-Tier System Architecture	29
4: <b>Figure 3.3.2.:</b> Entity Relationship Diagram for the OnTest System	31
5: <b>Figure 4.1.2:</b> Multi-step Quiz Creation Interface	36
6: <b>Figure 4.1.3:</b> Class detail page with enrolled students and assigned quizzes	37
7: <b>Figure 4.2.1:</b> Quiz interface with categorized listings	38
8: <b>Figure 4.2.2:</b> UI for MCQ question type	39
9: <b>Figure 4.3.2:</b> UI for Manual Grading	43
10: <b>Figure 4.3.3A:</b> Class Participation and Performance Analytics	44
11: <b>Figure 4.3.3B:</b> Comprehensive Analytics Dashboard – Teacher View	44

## **ABSTRACT**

In this project, a web-based platform for developing, administering, and evaluating online tests, OnTest, is designed and put into use. Multiple choice, true/false, matching, short answer, and essay questions are among the many kinds of questions supported by OnTest. It also offers capabilities for automated grading, quiz assignments that are given in class, and comprehensive performance statistics for both teachers and students. A dashboard for tracking student participation and learning objectives, a manual grading interface for essay replies, and real-time feedback are all features of the system. The report includes documentation of important implementation details, interface designs, and system procedures. Although the platform accomplishes the majority of its original objectives, there are still certain shortcomings in areas like LMS integration, test security, and semantic evaluation. These are prioritized for improvement in the future.



# **CHAPTER I**

## **INTRODUCTION**

### **1.1 Background and Motivation**

Assessment practices in education have changed dramatically as a result of the quick development of digital technologies. Online platforms that provide a variety of question formats, automatic scoring, and real-time data are gradually replacing traditional paper-based assessments. For example, Testportal offers an extensive online assessment tool with multiple-choice, matching, true/false, short answer, and essay question types, as well as full reporting tools. Similar to this, ClassMarker provides an adaptable online testing platform that allows teachers to design unique tests with features like time constraints, randomly generated questions, real-time feedback, and adjustable settings [1] [2].

The demand for data-driven insights and individualized learning is another factor propelling the move to online tests. As an example, Synap is a scalable and safe online test platform that allows teachers create personalized tests, do in-depth analysis, and interface with current systems. Additionally, studies have demonstrated that online tests can raise student involvement and motivation. According to a study done in Bulgaria during the COVID-19 pandemic, university students who used online learning platforms

instead of traditional ones showed higher levels of desire and a greater drive for achievement [3] [4].

Given these advancements, the necessity for an online platform that supports a variety of assessment formats and offers strong analytics, tailored feedback, and improved security features is increasing. A platform like OnTest would serve both teachers and students, facilitating performance monitoring, efficient practice, and ongoing development.

## **1.2 Problem Statement**

Traditional evaluation methods, which are largely manual in nature, present various obstacles in the ever-changing educational scene. Teachers frequently spend too much time writing and evaluating examinations, increasing their burden and delaying timely feedback for students. This delay can interfere with students' ability to learn and improve their performance.

With the growing trend towards digital learning environments, there is an urgent need for assessment systems that are efficient, scalable, and adaptive to a variety of learning scenarios. Current automated assessment platforms have made progress in resolving some of these challenges, but limitations remain. Many systems lack the flexibility to handle a wide range of question kinds, from multiple-choice to essay exercises, and frequently fail to give real-time, individualized feedback [5] [6].

The increasing need for individualized learning experiences highlights the importance of this issue. With technology and adaptive learning platforms, personalized learning maximizes educational resources and enables students to concentrate on areas that

require improvement, therefore speeding up their progress. Educational institutions run the danger of jeopardizing the integrity and quality of student evaluations if these issues are not resolved, which could influence learning outcomes and the efficacy of education as a whole.

The purpose of OnTest is to create and test an integrated assessment platform that tackles these issues. Designing a system that supports a variety of question formats: multiple-choice, matching, true/false, short answer, and essay question types, integrating cutting-edge analytics to offer real-time insights into student performance, ensuring a smooth integration with current class management tools. By fulfilling these goals, the study hopes to improve assessment practices and make sure they reflect the demands of modern education and technological developments.

### **1.3 Research Objectives**

This platform aims to satisfy the diverse assessment requirements across many subjects and educational levels, the platform will be developed to handle a broad range of question types, from simple multiple-choice to more advanced formats like essay exercises. The usefulness of providing a variety of question kinds, including multiple choice, fill-in-the-blanks, drag-and-drop, and audio/image-based questions, has been shown by platforms like Digiexam. This enables teachers to customize assessments to meet their unique educational objectives [7].

Additionally, in order to save teachers time and give students immediate feedback, the project intends to develop clever auto-grading algorithms that can assess

various question formats. Systems such as SpeedExam have demonstrated how real-time feedback and automated grading can improve student learning outcomes and teacher effectiveness [8].

Moreover, one of the main goals is to build a single educational environment that links assessment and class management tools, enabling teachers to manage students, design courses, assign tests, and monitor progress all from a single platform. Atomic Assessments and other platforms demonstrate how this connectivity improves efficiency and offers a smooth experience for both teachers and students [9].

Finally, the platform will concentrate on creating cutting-edge learning analytics tools to convert assessment results into useful information. By using these insights, educators can better understand the strengths and limitations of their students, modify their teaching methods, and promote individualized learning. Analytical data from automated grading systems has been demonstrated to forecast student performance and facilitate prompt interventions [10].

#### **1.4 Scope of Research**

The OnTest platform is designed to accommodate a wide range of assessment formats, including multiple-choice, matching, true/false, short answer, and essay questions. Diversifying assessment methods is critical for addressing varied learning styles and increasing student engagement. According to O'Neill and Padden (2021), diversified assessments can empower students and improve learning outcomes by giving them different ways to demonstrate mastery [11].

OnTest incorporates automated grading for objective question types to lessen the strain for teachers and give students instant feedback. It supports criteria-based grading systems for essay submissions, which expedite review. Studies indicate that automated evaluation greatly improves productivity and facilitates quicker learning cycles [12] [13].

The program enhances class management easier by allowing teachers to establish classes, offer tests, and measure progress all in one place. This integration lowers fragmentation while improving instructional and assessment continuity. Effective virtual class management is vital to sustaining student engagement and learning efficiency [14].

OnTest offers a learning analytics dashboard that shows student performance, progress, and trends over time. This enables teachers to make data-driven decisions and personalize instruction. Studies demonstrate that learning dashboards promote student engagement, self-regulation, and instructional efficacy [15].

The OnTest platform is intended to improve online educational exams by include features such as multiple question kinds, automatic grading, integrated class management, and learning analytics. However, certain capabilities are purposely removed from the current project scope in order to retain focus and assure timely delivery. These exclusions include the use of AI-based automatic grading for essay replies, integration with external Learning Management Systems (LMS), the creation of standalone mobile applications, and online proctoring using video surveillance.

## **1.5 Limitation**

OnTest provides a strong platform for online tests, however it currently has a number of issues that could affect its usability and efficacy. Because it lacks the ability to appropriately evaluate subjective replies like essays, the automated grading system is restricted to objective question categories and uses simple string matching for short-answer questions. The inability to assess inventiveness and nuanced comprehension in student responses is a typical shortcoming of automated grading systems [16].

Additionally, the platform's integration capabilities are limited; it is incompatible with well-known Learning Management Systems (LMS) such as Moodle and Canvas, and it does not provide APIs for smooth data interchange with institutional student information systems, which is a problem that frequently arises in LMS administration [17].

Another issue is accessibility, since OnTest is presently only accessible online and does not enable mobile devices or offline functioning, which may make it difficult for students with poor internet connectivity to use it.

Finally, the platform's learning analytics are simplistic and lack sophisticated capabilities like thorough behavioral analysis or AI-driven performance forecasts, which restricts teachers' capacity to customize lessons and interventions.

## **CHAPTER II**

### **LITERATURE REVIEW**

#### **2.1 Modern Educational Assessment Methods**

##### **2.1.1 Traditional vs Online Assessment**

Conventional tests, including paper exams, provide standardized evaluation techniques and controlled settings. Nonetheless, they frequently prioritize rote memory and might not adequately evaluate higher-order cognitive abilities. On the other hand, flexibility, instant feedback, and the opportunity to use a variety of question styles offered by online tests improve learning results and student engagement. Notwithstanding these benefits, online tests have drawbacks, such as maintaining academic integrity and resolving technical difficulties [18].

The advantages of online assessments are numerous and include less administrative labor, quicker grading, and instant feedback for students. Multimedia integration and a variety of learning schedules are supported by its adaptability, which raises accessibility and engagement. Because of these advantages, online evaluation is a potent instrument for improving the efficacy of instruction [19].

Despite its advantages, online assessments have drawbacks, such as concerns about accessibility, technological dependability, and maintaining academic integrity. Online tests are susceptible to cheating if they are not properly proctored, and system

malfunctions can ruin the learning experience. The validity of complicated tests may also be impacted when they are converted to digital formats [20] [21].

A balanced approach is offered by blended assessment, which combines traditional and online formats to combine efficiency with more in-depth face-to-face engagement. By enabling teachers to use a variety of assessment formats, this strategy improves flexibility and aligns instruction. It encourages critical thinking abilities while supporting differentiated training [22] [23].

### **2.1.2 Effective Purpose of Various Question Types**

To measure student comprehension effectively, assessment design must match question types to learning objectives. For assessing basic knowledge and comprehension across a wide range of subject areas, multiple-choice and true/false questions work effectively. Higher-order thinking abilities like analysis and synthesis, on the other hand, are better evaluated by short-answer and essay questions. In addition to efficiently measuring student learning, assessments that encourage deeper engagement with the content can be produced by teachers by carefully choosing and mixing various question styles. The validity and reliability of assessments are improved by this kind of strategic alignment, guaranteeing that they are useful instruments for assessment and education [24].

## **2.2 Existing Online Assessment Systems**

### **2.2.1 Analysis of Popular Platforms**

Popular game-based learning systems Kahoot and Quizizz increase student engagement by using interactive tests. While Quizizz offers both live and asynchronous



evaluations, giving classes more flexibility, Kahoot places more emphasis on real-time competition with live quizzes. Both platforms work well for formative examinations, but because they concentrate on multiple-choice questions, they might not be as good at assessing higher-order thinking abilities [25].

Both Moodle and Canvas are all-inclusive Learning Management Systems (LMS) that facilitate a variety of educational endeavors. Because of Moodle's great degree of flexibility and customization, teachers can adapt the platform to meet their unique demands. Canvas provides simplified course management tools and an easier-to-use interface. Both platforms, however, may have a high learning curve for novice users and necessitate a substantial time commitment to operate efficiently [26].

### **2.2.2 Market Gaps and Opportunities**

The market for online assessments is expected to develop rapidly due to technological advancements and the growing popularity of remote learning, from USD 1.69 billion in 2023 to USD 3.63 billion by 2032. Nevertheless, there is a need for solutions that are both adaptable and all-inclusive because many current platforms either only provide a small number of question kinds or lack sophisticated analytics. Additionally, there is a lack of flexible, real-time feedback mechanisms as the use of AI and immersive technologies in evaluations is still in its infancy. Platforms like OnTest have a great chance to meet the changing demands of contemporary education by offering a variety of question types and comprehensive analytics dashboards by addressing these unmet needs [27] [28].

## **2.3 Automated Grading Algorithms**

### **2.3.1 Automated Grading Accuracy**

Research shows that, depending on the type of question, automated grading systems can get accuracy rates of 85% to 98%. While short-answer and essay questions typically have accuracy rates between 80% and 92%, multiple-choice questions frequently have accuracy levels above 95%. Though they might have trouble with complex or context-dependent replies, these systems provide reliable assessments and lessen human biases [29].

### **2.3.2 OnTest's Method for Improving Accuracy**

OnTest focuses on objective question grading schemes that are based on rules rather than artificial intelligence, guaranteeing consistency and transparency. High accuracy levels are maintained by OnTest through the use of transparent grading rubrics and ongoing calibration against human grading standards. While reducing the issues with AI-driven grading systems, this method enables scalability and instant response [30].

### **2.3.3 Methods of Grading Various Question Types**

Techniques including string matching, keyword detection, and semantic similarity analysis are used in automated short-answer question grading. Although string matching is simple, it frequently ignores synonyms or other wording. Although keyword-based methods increase adaptability, they could still overlook subtle meanings. By assessing the contextual meaning of responses, semantic similarity techniques which make use of word embeddings and syntactic analysis prove to increase accuracy. Regular

expressions are frequently employed to account for variances in the right responses to matching and fill-in-the-blank questions. To increase grading accuracy across a variety of question formats, OnTest integrates several techniques: string matching, keyword identification, semantic analysis, and pattern recognition [31] [32].

## **2.4 Learning Analytics**

### **2.4.1 Key Metrics for Evaluating Learning Effectiveness**

Assessing the efficiency of learning entails monitoring key performance indicators (KPIs) such test results, time spent on task, accuracy rates, and student development over time. While assessment results offer a clear indicator of knowledge acquisition, student effort and engagement are reflected in time spent on task. Monitoring progress over time enables teachers to evaluate growth and modify training accordingly, and accuracy rates assist in identifying areas where children might want more attention. All of these measures work together to provide a thorough understanding of student learning and support data-driven decision-making in learning environments [33].

### **2.4.2 Using Data to Improve Learning Outcomes**

By determining the strengths and opportunities for growth of each student, learning analytics enables teachers to customize their instruction. Teachers can adapt curriculum to better suit the requirements of their students, apply focused interventions, and personalize learning experiences by examining data such as assessment results, engagement indicators, and learning behaviors. When students get access to their own learning data, it encourages introspection and strategic planning, which helps them develop

better study habits and ask for help when they need it. This process is made easier by platforms such as OnTest, which offer user-friendly dashboards with real-time statistics to help teachers and students make well-informed decisions that will improve learning outcomes [34].

## CHAPTER III

### METHODOLOGY

#### 3.1 Development Process

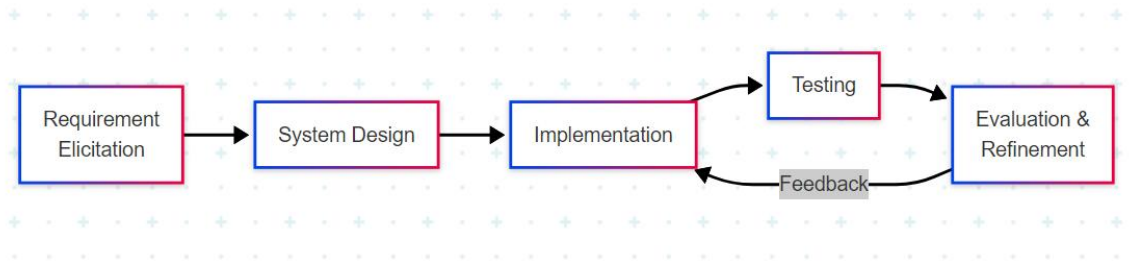
**Iterative and Incremental Development Model:** OnTest adopted an Agile-inspired iterative methodology, prioritizing incremental delivery and adaptive planning to navigate the evolving requirements of educational technology. Four key iterations were executed, each spanning 3–4 weeks and focusing on *vertical slices* of functionality (e.g., authentication flows, quiz creation modules). Feedback from stakeholders, including academic advisors and pilot users - was systematically integrated, embodying the Agile Manifesto’s principle of "responding to change over following a rigid plan".

**Core Iterative Cycles:** The development process unfolded through overlapping phases, as illustrated in Figure 3.1:

**Table 3.1: Iteration Breakdown and Major Activities**

Iteration	Objective	Activities
Iteration 1: Core Infrastructure	Establish foundational architecture and user authentication.	<ul style="list-style-type: none"><li>- Designed PostgreSQL schemas for User and Class models using Prisma ORM.</li><li>- Implemented NextAuth.js with JWT sessions.</li><li>- Conducted security audits using SonarQube.</li></ul>
Iteration 2: Assessment Engine	Develop question bank and quiz creation workflows.	<ul style="list-style-type: none"><li>- Built reusable React components (QuestionEditor, QuizConfigurator) with TypeScript interfaces.</li></ul>

Iteration	Objective	Activities
		<ul style="list-style-type: none"> <li>- Integrated Tiptap for rich-text question formatting, supporting LaTeX equations and code blocks.</li> <li>- Validated UI workflows.</li> </ul>
Iteration 3: Analytics and Reporting	Enable data-driven decision-making for educators.	- Developed ScoreAnalytics and ClassPerformance models with Chart.js visualizations.
Iteration 4: Polishing and Accessibility	Refine UX and ensure inclusivity.	- Added keyboard navigation for timed assessments, improving screen reader compatibility.



**Figure 3.1:** Iterative Development Cycle

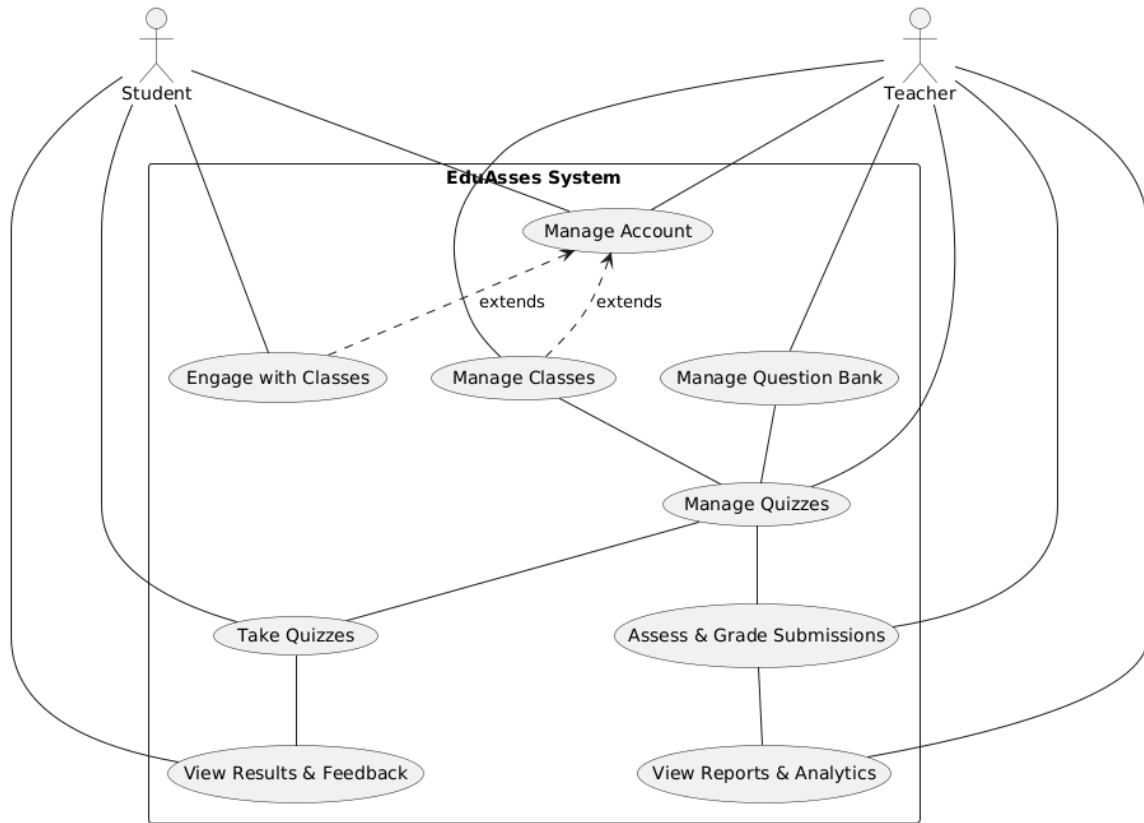
### 3.2 System Requirements Analysis

This section details the systematic analysis undertaken to define the requirements for the OnTest system. The primary objective of this analysis is to clearly articulate the functionalities the system must provide and the quality attributes it must exhibit. This comprehensive understanding of requirements serves as a critical foundation for the subsequent design, development, and evaluation phases of the project, ensuring that the final system aligns with its intended goals and user expectations.

### **3.2.1 Functional Requirements**

Functional requirements define the specific behaviors, tasks, and operations that the OnTest system must be capable of performing. These requirements are derived from the core objectives of the project, which is to develop an effective and user-friendly online assessment platform tailored for educational environments. They specify what the system does.

To provide a high-level overview of the primary interactions between users and the OnTest system, a Use Case diagram is presented below (Figure 3.2.1). This diagram illustrates the main actors (user roles) and the key functionalities (use cases) they engage with to achieve their objectives within the system.



**Figure 3.2.1:** Use Case Diagram of the OnTest System

The Use Case Diagram (Figure 3.2.1) illustrates the primary interactions between the identified actors - Teacher and Student - and the core functionalities of the OnTest system. Each use case represents a significant goal that an actor can achieve using the system. Table 3.2.1 provides a detailed description of these use cases.

**Table 3.2.1: Description of Use Cases in the OnTest System**



<b>Use Case ID</b>	<b>Use Case Name</b>	<b>Primary Actor(s)</b>	<b>Description</b>
UC1	Manage Account	Teacher, Student	Allows users to register for a new account, log in, log out, and manage their personal profile information.
UC2	Manage Classes	Teacher	Enables teachers to create new classes, edit existing class details, manage student enrollment, and delete classes.
UC3	Manage Question Bank	Teacher	Facilitates the creation, organization, editing, and deletion of assessment questions in a reusable repository.
UC4	Manage Quizzes	Teacher	Allows teachers to design new quizzes, add questions, configure settings (e.g., time limits, attempts), and publish them to classes.
UC5	Assess and Grade Submissions	Teacher	Enables teachers to review student quiz attempts, provide grades for subjective questions, and offer feedback.
UC6	View Reports and Analytics	Teacher	Provides teachers with access to performance data, class statistics, and analytical insights from quiz results.
UC7	Engage with Classes	Student	Allows students to join classes using a code, view class announcements, and access shared learning materials.
UC8	Take Quizzes	Student	Enables students to undertake assigned or available quizzes, submit their answers within the defined constraints.
UC9	View Results and Feedback	Student	Allows students to review their performance on completed quizzes, including scores and teacher feedback.

These use cases collectively define the core functionalities of the OnTest System. Each use case represents a key interaction between the system and its users, ensuring that both teachers and students can perform their essential tasks efficiently. The

descriptions provide a high-level overview, while detailed system behaviors and interaction scenarios are elaborated in the following Functional Requirements Specification section. This structured use-case model facilitates streamlined development, testing, and future expansion of system capabilities.

The functional requirements Specification provides a comprehensive and granular specification of the functional capabilities that the OnTest system must deliver. Building upon the high-level interactions depicted in the Use Case Diagram (Figure 3.2.1), these detailed Functional Requirements (FRs) articulate the precise actions, system behaviors, and data manipulations necessary to fulfill the objectives of each use case.

Each functional requirement is uniquely identified, traceable to one or more use cases, and expressed in a way that supports clarity for system design, implementation, and verification. The goal is to translate abstract user interactions into actionable, testable, and implementable functionality.

The requirements are categorized by user roles, providing an organized structure that reflects the distinct operational needs and permissions of different stakeholders within the OnTest ecosystem.

#### **A. User Roles**

The OnTest system is architected around two principal user roles, each playing a crucial role in the educational assessment lifecycle. These roles are explicitly defined to guide system design choices, enforce access control policies, and streamline the user experience.

Teacher: The Teacher role encompasses the responsibilities of content creation, quiz configuration, class management, and performance evaluation. Teachers are empowered to construct assessments, manage question banks, monitor student activity, and interpret analytical insights derived from student submissions.

Student: The Student role focuses on interaction with the learning content and assessments prepared by Teachers. Students can enroll in classes, take quizzes, and receive personalized feedback and analytics to support their academic development.

## **B. Requirements for Teacher Role**

The Teacher role constitutes the pedagogical backbone of the OnTest system. It encompasses all actions related to content creation, assessment configuration, student engagement, and performance evaluation. This section defines the specific system behaviors and capabilities required to support teachers throughout the full online assessment lifecycle. The functional requirements are categorized into six main functional clusters, each representing a coherent group of features tailored to this user role. Each requirement is presented with a unique identifier (FR-TXX) to support traceability throughout the development, testing, and validation phases of the project.

### **B1. Authentication and Authorization**

To ensure secure and role-based access, the system must provide robust authentication and identity management mechanisms for teachers. These features prevent unauthorized access and uphold the integrity of sensitive educational data.

**Table 3.2.1.B1 – Authentication and Authorization Requirements**

ID	Requirement Description
FR-T01	The system shall allow teachers to register for a new account using their email and a password.
FR-T02	The system shall enable teachers to securely log in using their credentials via NextAuth.js.
FR-T03	The system shall store passwords in a hashed format, not plaintext, using industry standards.
FR-T04	The system shall maintain active sessions to prevent frequent re-authentication during use.
FR-T05	Teachers shall be able to update their profile information (e.g., name, contact details).
FR-T06	Teachers shall be able to log out securely, ending their session and access to the system.

## **B2. Class Management**

To support a structured teaching environment, the platform must provide teachers with tools to create, organize, and oversee their courses and students.

**Table 3.2.1.B2 – Class Management Requirements**

ID	Requirement Description
FR-T07	Teachers shall be able to create new classes with metadata such as name, description, subject.
FR-T08	Teachers shall view all their created classes on a personalized dashboard.
FR-T09	Teachers shall be able to edit existing class details.
FR-T10	Teachers shall be able to delete classes with confirmation prompts regarding data impact.
FR-T11	The system shall generate a unique class join code for student self-enrollment.
FR-T12	Teachers shall view enrolled student lists per class.
FR-T13	Teachers shall post announcements to classes via the ClassAnnouncement model.

ID	Requirement Description
FR-T14	Teachers shall upload and share learning materials to classes using the Material model.

### B3. Question Bank Management

Teachers must be able to manage a personal repository of questions to streamline the quiz creation process and promote content reusability.

**Table 3.2.1.B3 – Question Bank Management Requirements**

ID	Requirement Description
FR-T15	Teachers shall create questions within a personal/institutional question bank.
FR-T16	The system shall support multiple question types: MCQ, True/False, Fill in the Blank, Short Answer, and Essay.
FR-T17	Teachers shall define answer options and correct responses for objective-type questions.
FR-T18	Teachers shall assign default point values per question.
FR-T19	Teachers shall be able to view, edit, and delete questions from their bank.
FR-T20	Teachers shall categorize questions by topic, difficulty, or learning objective. <i>(optional)</i>

### B4. Quiz Management

This functionality allows teachers to compose assessments using the question bank, configure quiz parameters, and control availability.

**Table 3.2.1.B4 – Quiz Management Requirements**

ID	Requirement Description
FR-T21	Teachers shall create quizzes with title, description, and instructions.

ID	Requirement Description
FR-T22	Teachers shall add questions from their bank or create new ones during quiz creation.
FR-T23	Teachers shall define or randomize question ordering.
FR-T24	Teachers shall configure quiz settings including duration, attempt limits, availability window, answer review rules, and randomization settings.
FR-T25	Teachers shall assign quizzes to specific classes.
FR-T26	Teachers shall publish or unpublish quizzes as needed.
FR-T27	Teachers shall view a list of all quizzes with their status (e.g., draft, published).
FR-T28	Teachers shall preview quizzes as students before publishing.

### **B5. Assessment and Grading**

OnTest must facilitate both automated and manual grading workflows, providing tools for personalized evaluation and feedback.

**Table 3.2.1.B5 – Assessment and Grading Requirements**

ID	Requirement Description
FR-T29	Teachers shall track student submission status via the QuizAttempt model.
FR-T30	The system shall auto-grade objective question types based on predefined answers.
FR-T31	Teachers shall manually grade subjective answers (Essay, Short Answer) with comment support.
FR-T32	Teachers shall record and finalize total quiz scores per student.
FR-T33	Teachers shall provide overall feedback per quiz attempt in addition to per-question comments.

### **B6. Reporting and Analytics**

To support data-driven instruction, the system must deliver meaningful performance metrics and insights into student learning outcomes.

**Table 3.2.1.B6 – Reporting and Analytics Requirements**

ID	Requirement Description
FR-T34	Teachers shall view class-wide aggregated scores and statistics per quiz (e.g., average, median).
FR-T35	Teachers shall access individual quiz attempts and student responses.
FR-T36	The system shall provide analytical insights (e.g., score distributions, question success rates) via models such as ScoreAnalytics, ClassPerformance.

This structured presentation of the functional requirements for the **Teacher role** ensures clarity, coverage, and a solid foundation for both system implementation and verification phases. It directly aligns with the objectives identified in the system’s use case model and prioritizes usability, security, and pedagogical effectiveness.

### **C. Requirements for Student Role**

The Student role is dedicated to learning participation, class interaction, and assessment submission within the OnTests platform. The system must provide a user-friendly and responsive environment to facilitate student activities, including secure access, quiz taking, and receiving feedback.

#### **C1. Authentication and Authorization**

Students must be able to securely access the platform to engage in assessments and learning activities. Ensuring data protection and ease of use is vital for a seamless student experience.

**Table 3.2.1.C1 – Reporting and Analytics Requirements**

<b>ID</b>	<b>Functional Requirement</b>
FR-S01	The system shall allow students to register for a new account, typically using an email address and password.
FR-S02	The system shall enable registered students to log in securely using their credentials.
FR-S03	The system shall ensure student passwords are securely hashed before storage.
FR-S04	Students shall be able to update their personal profile information.
FR-S05	Students shall be able to log out of the system securely.

### **C2. Class Engagement**

To participate in learning activities, students must be able to join classes, receive announcements, and access course materials uploaded by their teachers.

**Table 3.2.1.C2 – Class Engagement Requirements**

<b>ID</b>	<b>Functional Requirement</b>
FR-S06	Students shall be able to join a class by entering a unique join code provided by their teacher.
FR-S07	Students shall be able to view a list of all classes they are enrolled in via their personalized dashboard.
FR-S08	Students shall be able to view class announcements and access learning materials shared by the teacher within each enrolled class.

### **C3. Quiz Taking**

The system must provide a robust and intuitive interface for students to take quizzes efficiently and reliably.

**Table 3.2.1.C3 – Quiz Taking Requirements**



ID	Functional Requirement
FR-S09	Students shall be able to view a list of quizzes assigned to them or currently available, with details such as due dates and time limits.
FR-S10	Students shall be able to initiate a quiz attempt if the quiz is active and within allowed attempt limits.
FR-S11	The quiz interface shall display the question content, any associated media, and proper input fields for answering.
FR-S12	Students shall be able to input or select their answers depending on the question type.
FR-S13	If a quiz has a time limit, a visible countdown timer shall be displayed.
FR-S14	Students shall be able to navigate freely between questions during a quiz attempt.
FR-S15	The system should auto-save student responses periodically to minimize data loss due to unexpected disruptions.
FR-S16	Students shall be able to submit their quiz for grading, with automatic submission if the time limit expires.

#### C4. Results and Feedback

Feedback and performance insights help students reflect on their progress and understand areas of improvement.

**Table 3.2.1.C4 – Reporting and Analytics Requirements**

ID	Functional Requirement
FR-S17	Upon completion of grading (manual or automatic), students shall be able to view their scores for submitted quizzes.
FR-S18	Based on quiz settings, students may review their submitted answers, see correct answers (for objective questions), and receive feedback from teachers.
FR-S19	Students shall have access to a history of their completed quizzes, including scores and feedback, for future reference.

### 3.2.2. Non-functional Requirements

Non-functional requirements (NFRs) define the quality attributes, performance standards, and other system constraints that do not directly relate to specific functionalities but are essential for successful user experience and overall system reliability.

**Table 3.2.2 – Non-functional Requirements**

ID	Requirement Title	Detailed Description / Acceptance Criteria
NFR01	Usability	<ul style="list-style-type: none"><li>- The user interface must be intuitive and user-friendly for both teachers and students, including users with limited technical experience.</li><li>- Workflows for key tasks (e.g., creating quizzes, taking tests) should be logical and minimize unnecessary steps.</li><li>- The system must provide clear feedback for user actions (e.g., success or error messages).</li></ul>
NFR02	Performance	<ul style="list-style-type: none"><li>- Page load time for core pages (dashboard, quiz list, test interface) must not exceed 3–5 seconds under normal network conditions.</li><li>- The system must respond to user interactions (e.g., saving answers, submitting quizzes) within an acceptable delay (ideally &lt; 300ms).</li><li>- The system should support a reasonable number of concurrent users (e.g., 50–100 students) without significant performance degradation.</li></ul>
NFR03	Reliability	<ul style="list-style-type: none"><li>- Student responses must be stored accurately and completely.</li><li>- The system should function stably with minimal unexpected runtime errors.</li><li>- Automated grading must follow the defined answer key and logic precisely.</li></ul>
NFR04	Security	<ul style="list-style-type: none"><li>- User authentication must be securely implemented (e.g., using NextAuth.js with hashed passwords).</li><li>- Sensitive data (e.g., personal information, test content, grades) must be protected from unauthorized access.</li><li>- Role-based access control (RBAC) must be strictly enforced: teachers can only access data relevant to their own classes, and students can only view their own information.</li></ul>

ID	Requirement Title	Detailed Description / Acceptance Criteria
NFR05	Scalability	<ul style="list-style-type: none"> <li>- The system architecture and database design (using Prisma) should support moderate scaling to accommodate increasing users and data over time (e.g., across multiple classes or school terms).</li> <li>- Deployment via platforms such as Vercel should support scaling under the project scope.</li> </ul>
NFR06	Maintainability	<ul style="list-style-type: none"> <li>- Source code must be well-organized into functional modules.</li> <li>- Naming conventions for variables, functions, and components should be consistent and meaningful.</li> <li>- React components should be designed for reusability.</li> <li>- Tight coupling between modules should be minimized.</li> </ul>
NFR07	Accessibility	<ul style="list-style-type: none"> <li>- Users must be able to navigate and perform core actions via keyboard.</li> <li>- Important images must include alt text.</li> <li>- Sufficient color contrast must be maintained for visually impaired users.</li> </ul>
NFR08	Responsiveness	<ul style="list-style-type: none"> <li>- The UI must display and function well across a variety of screen sizes, including desktops, tablets, and smartphones.</li> <li>- A mobile-first design approach should be applied.</li> <li>- Layouts must remain intact and functional on common device resolutions.</li> </ul>

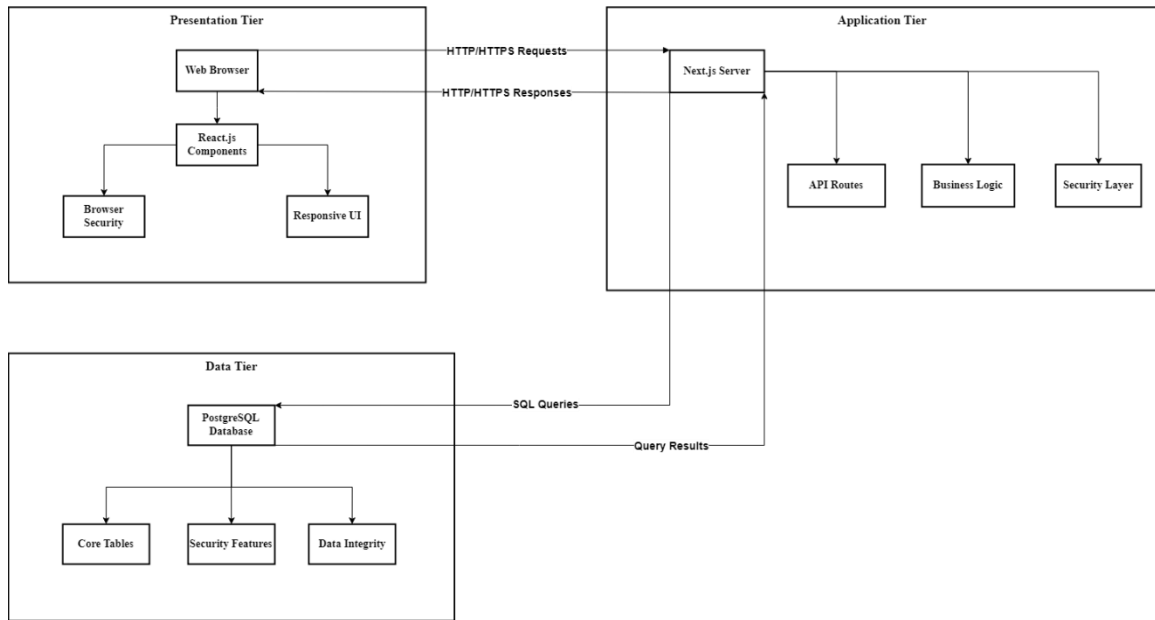
### 3.3 System and Database Design

This section outlines the integrated design of the OnTest system, detailing its software architecture and the underlying database structure. The objective is to provide a cohesive understanding of how the system is constructed and how data is managed to support its diverse functionalities, presented in a concise manner that emphasizes core design decisions.

### 3.3.1 System Architecture

The OnTest system is architected as a modern client-server application, leveraging the Next.js 15 App Router to provide a full-stack development environment. This architecture is centered around React 19 for building a dynamic and interactive user interface.

**Architectural Pattern Overview:** Figure 3.3.1 illustrates the principal components of the OnTest architecture and their interactions that the client-server model forms the basis of the system, where the user's browser (client) interacts with the Next.js backend. A key characteristic of this architecture within OnTest is the strategic utilization of React Server Components (RSCs) for efficient data fetching and rendering on the server, complemented by Client Components that manage user interactivity and client-side state. This hybrid approach aims to optimize both initial page load performance and the responsiveness of the user interface. API Route Handlers within the Next.js framework serve as the backend endpoints, processing client requests and orchestrating business logic.



**Figure 3.3.1: 3-Tier System Architecture**

**Key System Modules and Their Interactions:** The OnTest system is logically segmented into core functional modules. This modularity enhances organization and maintainability. The primary modules include:

- **Authentication Module:** Governs user sign-up, login, and session control, utilizing NextAuth.js to secure system access and manage user identities across all other modules.
- **User and Class Management Module:** Manages user profiles and facilitates the creation and administration of classes, including student enrollment and the distribution of learning materials, interacting heavily with the database via Prisma.

- **Question and Quiz Management Module:** Enables teachers to develop and organize a diverse range of questions within a structured question bank and assemble these into configurable quizzes, defining the core assessment content.
- **Quiz Taking and Submission Module:** Provides the interactive interface for students to undertake quizzes, submit their responses, and manages the lifecycle of a quiz attempt.
- **Grading and Results Module:** Supports teachers in the evaluation of student submissions and enables both teachers and students to view performance outcomes and feedback, thereby completing the assessment loop.

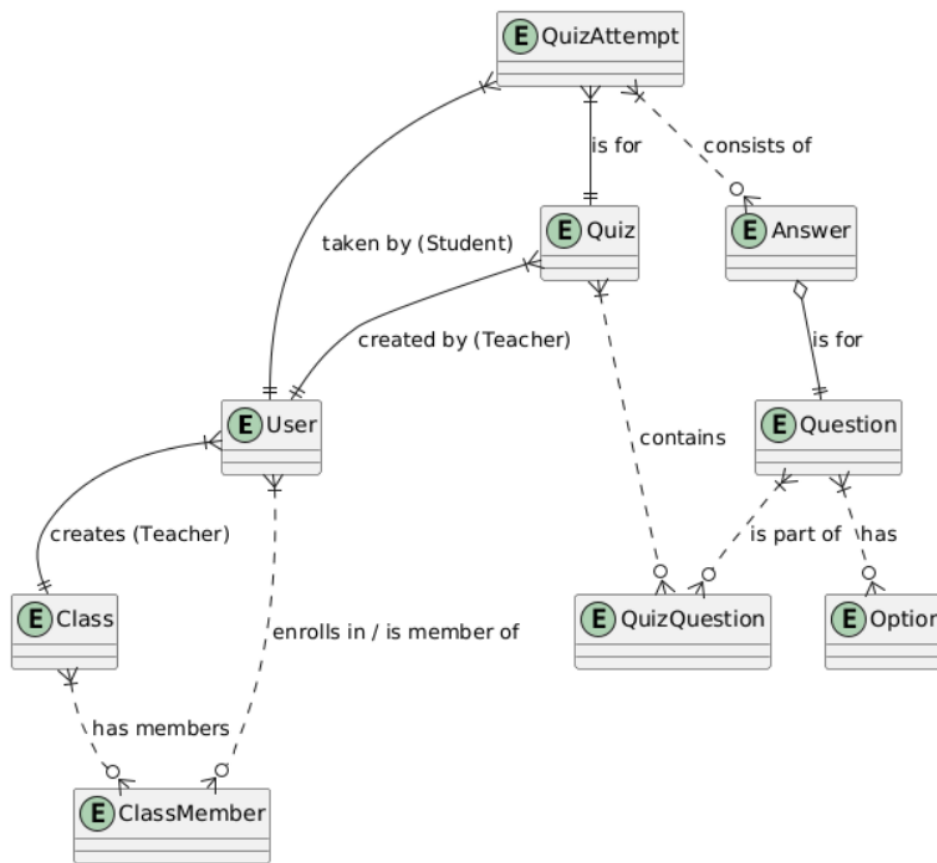
Each module relies on the underlying Next.js framework for routing and API handling, Prisma for database operations, and NextAuth.js for ensuring secure, role-appropriate access to its functionalities.

### **3.3.2 Database Design**

The foundation of OnTest's data management strategy is its relational database design, implemented in PostgreSQL and accessed via the Prisma ORM. This section focuses on the core data model and how data flows through key entities during system operation.

**Data Model and Entity Relationship Diagram (ERD):** The integrity and functionality of the OnTest system are heavily reliant on a well-defined data model. This model captures all essential information entities and the relationships between them. The

Entity Relationship Diagram (ERD) presented in Figure 3.Z visually represents this data structure, as defined and managed by the Prisma schema within the OnTest project. It serves as the blueprint for how information regarding users, educational content, assessments, and interactions is stored and interconnected.



**Figure 3.3.2.:** Entity Relationship Diagram for the OnTest System

Analysis of Key Entities and Relationships in OnTest: The ERD (Figure 3.3.2) illustrates the various entities and their interconnections. The following key entities and their relationships are fundamental to the OnTest system's operation:

**Table 3.3.2: Analysis of Key Entities and Their Relationships**

Entity	Relationship / Description
User	- Central entity representing system users (Teachers, Students). - Stores name, email, hashed password, and role. - Role defines permissions and access scope.
	- A Teacher User can CREATE multiple Class (1:N).
	- A Teacher User can CREATE multiple QuestionBank and Quiz (1:N).
	- A Student User can ENROLL_IN multiple Class entities via ClassMember (M:N).
	- A Student User can HAVE multiple QuizAttempt records (1:N).
Class	- Represents a virtual class.
	- Each Class BELONGS_TO one User (Teacher) (N:1).
	- A Class HAS multiple enrolled Users (Students) via ClassMember (M:N).
	- A Class HAS multiple ClassAnnouncement and Material entities (1:N).
	- A Class can be ASSIGNED multiple Quiz entities. Depending on system design: <ul style="list-style-type: none"> <li>• If a quiz belongs to one class → 1:N</li> <li>• If reusable → M:N with junction.</li> </ul>
Quiz	- Assessment entity.
	- Each Quiz is CREATED_BY one User (Teacher) (N:1).
	- A Quiz CONTAINS multiple Question entities via QuizQuestion (M:N or 1:N depending on reusability).
	- A Quiz can HAVE multiple QuizAttempt records (1:N).
Question	- Represents an assessment item.
	- Has a type (e.g., MULTIPLE_CHOICE, ESSAY).
	- A multiple-choice Question HAS_MANY Option entities (1:N).
	- A Question can BELONG_TO one or more QuestionBank entities (1:N or M:N).
Option	- Stores answer choices for multiple-choice questions.
	- Each Option BELONGS_TO one Question (N:1).
QuizAttempt	- Represents a student's submission.
	- Each QuizAttempt BELONGS_TO one User (Student) (N:1).
	- Each QuizAttempt is FOR one Quiz (N:1).



Entity	Relationship / Description
	- Each QuizAttempt CONSISTS_OF multiple Answer records (1:N).
	- Stores score, time range, and status.
Answer	- Represents an individual response within an attempt.
	- Each Answer BELONGS_TO one QuizAttempt (N:1).
	- Each Answer CORRESPONDS_TO one Question (N:1).
	- Stores answer content, score, and feedback.

The relationships, such as one-to-many and many-to-many, ensure data integrity and enable complex queries necessary for the system's functionalities, such as retrieving all questions for a quiz, all attempts by a student, or all students in a class.

## **CHAPTER IV**

### **IMPLEMENT**

#### **4.1 Assessment Creation and Management**

The OnTest system has been successfully developed as an online educational platform with a primary focus on creating a safe, wholesome learning environment that facilitates student engagement and connection. Built on Next.js 15 App Router architecture, the system integrates React, Prisma ORM with PostgreSQL, and NextAuth.js to deliver a secure yet accessible assessment platform. This technological foundation was specifically chosen to support a responsive, user-friendly experience that prioritizes educational interaction rather than mere assessment administration.

##### **4.1.1 Comprehensive Question Type Support**

To meet various assessment requirements, OnTest offers five fundamental question kinds, each with its own data format and user interface. With the use of a metadata element in the question model that specifies scoring behavior, multiple choice questions (MCQ) give teachers the flexibility to design questions with one or more right answers. Derived from the MCQ structure, True/False Questions provide an easy-to-use, simplified binary format with a simple editor. In order to provide accurate answer tracking and automated scoring for matching questions, the system pairs items using group and matchId fields within the Option model. Multiple valid answers are supported by short answer questions, which also allow for brief written responses and an optional casesensitive flag

to ensure accurate grading. Last but not least, essay questions encourage lengthy, open-ended answers and are assessed manually by teachers using a grading system that lets them give grades and provide comments. This modular question structure offers a scalable and adaptable basis for a variety of evaluation situations.

The image shows a web-based interface for creating a new multiple-choice question. At the top, there's a header 'New Multiple Choice Question' with 'Cancel' and 'Save Question' buttons. Below this is a tabbed interface with 'Question Content' and 'Settings' tabs. The 'Question Content' tab is active, showing a rich text editor with a toolbar (B, I, H1, H2, bulleted list, numbered list, link) and a large text area. Below the text area is the 'Answer Options' section, which has a toggle for 'Allow multiple correct answers' and four option rows. Each row contains a radio button, a text input field, and a delete icon. At the bottom of the 'Answer Options' section is an 'Add Option' button.

**Figure 4.1:** *Question Editor Interface for Multiple Question Types (MCQ).*

#### 4.1.2 Quiz Creation Workflow

OnTest employs a methodical, multi-phase process for creating quizzes that is tailored for both user experience and instructional adaptability. Instructors supply the title, subject, description, and tags, among other crucial quiz elements, in Step 1-Basic Information. The form interface's real-time validation and simple layout are intended to improve usability. Key factors like timeLimit, passingScore, maxAttempts, availability

period (startDate, endDate), result visibility (showResults), and question shuffling (shuffleQuestions) can all be customized in Step 2-Quiz Configuration. Sliders, date pickers, and toggle switches are examples of interactive user interface elements that offer a smooth setting experience. Teachers have three options for adding questions in Step 3-Question Management: they can create new ones, choose from an existing question bank, or import questions in bulk from a template. Live previews, drag-and-drop reordering, and point modification per question are all supported by the system. Lastly, before saving the quiz as a draft or publishing it live, Step 4-Preview and Publish provides a preview that is visible to students. This methodical procedure guarantees efficiency and control in the creation of quizzes.

The screenshot displays the 'Questions' tab of a quiz creation interface. At the top, the quiz title 'Software Design Principles (SOLID)' is shown, along with 'Preview' and 'Save' buttons. Below the title, there are three tabs: 'Details', 'Questions' (which is active), and 'Settings'. A '+ Add Question' button is located on the right. The main area is titled 'Quiz Questions' and includes a subtitle 'Drag to reorder questions. Click edit to modify a question.' It contains a list of five questions, each with a drag handle (three vertical dots), a question ID, type, points, and a description. Each question also has an edit icon (pencil) and a delete icon (trash can).

Question ID	Type	Points	Description
Q1	Multiple Choice	10 points	Câu hỏi trắc nghiệm 1 cho Software Design Principles (SOLID)?
Q2	True/False	5 points	Câu hỏi Đúng/Sai cho Software Design Principles (SOLID): Phát biểu này có đúng không?
Q3	Fill in the Blank	8 points	Câu hỏi điền vào chỗ trống cho Software Design Principles (SOLID): Từ còn thiếu là [blank].
Q4	Short Answer	12 points	Câu hỏi tự luận ngắn cho Software Design Principles (SOLID): Hãy giải thích khái niệm X.
Q5	Multiple Choice	15 points	Câu hỏi trắc nghiệm 2 (chọn nhiều đáp án) cho Software Design Principles (SOLID)?

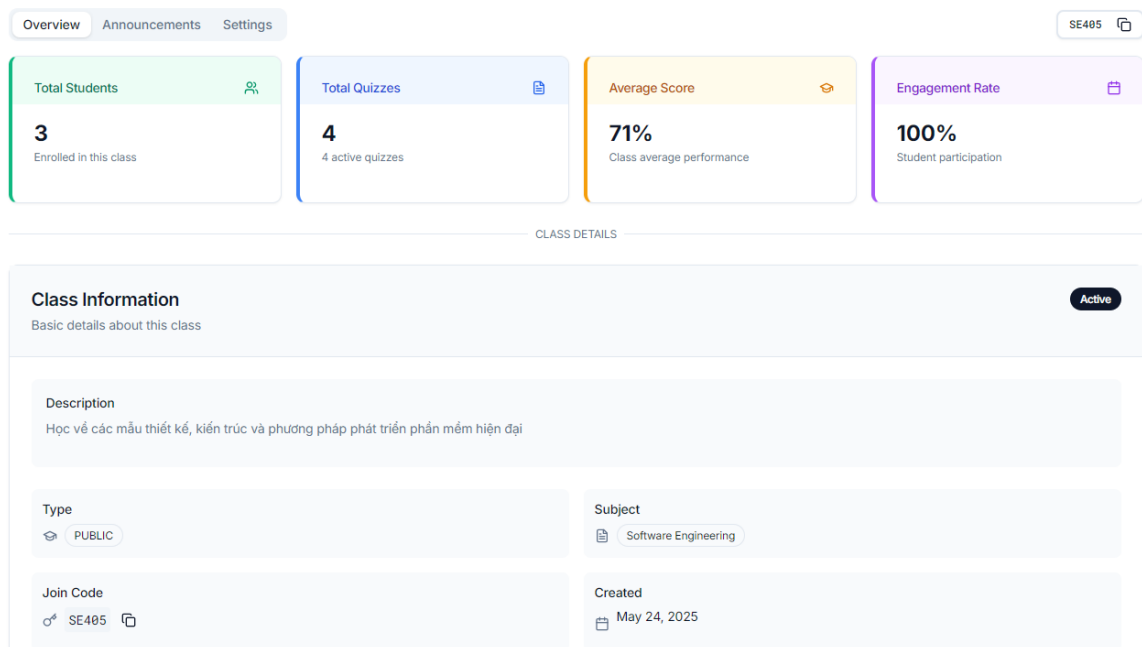
**Figure 4.1.2: Multi-step Quiz Creation Interface**

### 4.1.3 Class Management Integration

Flexible student registration is made possible by OnTest' support for both public and private classes. Using the ClassEnrollment paradigm, teachers can add or delete students from student lists using an intuitive interface. By using the classId field to link quizzes to classes, automatic visibility and alerts are ensured. Teachers can assist students stay informed about deadlines and developments by using an announcement system called ClassAnnouncement.

#### Kiến Trúc Phần Mềm

Học về các mẫu thiết kế, kiến trúc và phương pháp phát triển phần mềm hiện đại



**Figure 4.1.3:** Class detail page with enrolled students and assigned quizzes

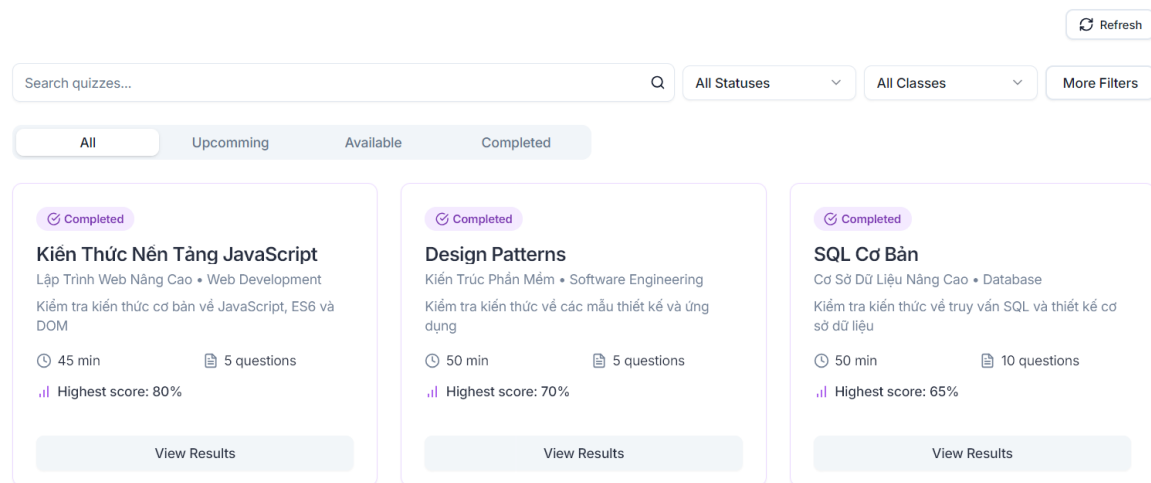
## 4.2 Test Participation and Submission Handling

### 4.2.1 Student Quiz Interface Design

With the help of colored cards, OnTest offers a simplified and user-friendly quiz interface that groups quizzes into four categories: upcoming, available, completed, and expired. A clear presentation of the questions, along with support for media, scoring, and type-specific interactions, is provided during an active attempt. Students use a quick-jump menu, progress meter, and simple buttons to navigate. Every question's current status—answered, unanswered, or flagged—is displayed. Students can monitor the amount of time left and prevent late submissions by using the always-visible countdown meter.

#### My Quizzes

View and take your assigned quizzes



*Figure 4.2.1: Quiz interface with categorized listings*

## 4.2.2 Question Type Interaction Implementations

In OnTest, every question type has a customized interface design that is suited for its format. In multiple choice and true/false questions, radio buttons are used for single responses and checkboxes for numerous answers. Keyboard shortcuts and clicks are supported. Dropdown menus are used in matching questions to align paired items, and a special data schema is used to monitor selections. Text input fields with real-time validation and optional tips are used in Short Answer. A rich text editor with word counts, configurable word limits, and basic formatting is included in essay questions.

### React Fundamentals

Kiểm tra hiểu biết về React hooks, components và JSX

⌚ 00:59:36

Câu 1/8

Hook nào được sử dụng để lưu trữ và cập nhật trạng thái trong React functional components?

✓ Đơn lựa chọn (Chỉ chọn một đáp án)

☐ useEffect

☐ useState

☐ useContext

☐ useReducer

Câu trước

Câu tiếp theo

Câu hỏi

Tất cả 8 | Đã làm 0 | Chưa làm 8 | Đánh dấu 0

1 2 3 4 5

6 7 8

Nộp bài

*Figure 4.2.2: UI for MCQ question type*

## 4.2.3 Submission Process and Integrity Assurance

A safe and dependable quiz submission process is guaranteed by OnTest. The "Submit" option allows students to manually submit their answers, or the system will

automatically submit them when the time limit has passed. Unintentional submissions are avoided using a confirmation dialog. Before completing the submission, the backend enforces all restrictions, including time limits and attempt counts. Grading starts after `QuizAttempt.completedAt` is recorded upon completion. Students are taken to their results if `showResults` is enabled; if not, they are given a success confirmation. Server-side checks verify time and attempt restrictions to ensure integrity, and auditing logs of metadata such as `ipAddress`, `userAgent`, and `timeSpent` are kept.

## **4.3 Automated Grading and Manual Grading**

### **4.3.1 Grading Algorithm Implementation**

In order to manage the core logic for multiple choice (MCQ) and true/false questions, OnTest uses the `gradeMultipleChoiceQuestion` function, which internally calls `gradeMultipleChoiceAnswer`. Through metadata (`allowMultiple`), the system differentiates between single-answer and multiple-answer forms. It also accommodates answer penalties and partial credit (`allowPartialCredit`). The student can only select one right answer in single-answer mode, and they only get full points if their decision is the right one. The algorithm assesses how many choices are right and wrong for multiple-choice questions. Full points are awarded to the student if all of the right answers are chosen and no wrong ones. A proportionate score is determined by deducting penalties for incorrect responses from the number of successfully chosen answers when partial credit is permitted. A numerical score, `isCorrect`, and optional commentary are all included in the outcome, which is always consistent with the established grading guidelines.



In order to evaluate short answer questions, the `gradeShortAnswerQuestion` function compares the student's input with a list of accepted, accurate answers. The question metadata's `caseSensitive` flag regulates case sensitivity, and each right response is represented as an `Option` with `isCorrect = true`. Through the removal of whitespace and optional conversion to lowercase, the method normalizes both the student's response and the predetermined correct responses. If any of the answers are correct, the student obtains full points; if not, they get a zero. The system marks a question for manual grading if there are no right responses or if the response is blank. This method strikes a mix between flexibility and automation, guaranteeing simple grading while permitting more intricate management when required.

The `gradeMatchingQuestion` function is used to grade matching-type questions by comparing student-submitted pairings to a list of predetermined correct matches. The question labels each option with a category (such as "premise" or "response") and a `matchId` that associates it with a matching pair. The response from the student is saved as a JSON object that maps the IDs of the premise options to the IDs of the chosen response options. This structure is parsed by the algorithm, which then verifies each match. The ultimate score, which is calculated as the ratio of right pairings to the total number of projected matches, is influenced by a correct match. Partial credit is automatically given based on the quantity of accurate links, while full credit is only given if every pair is correct. Instant feedback from the system, such as "3/5 correct matches," enables students to properly and simply comprehend their performance.

### 4.3.2 Manual Grading Interface for Essays

For essay-style questions, OnTest offers a simplified manual grading interface that ensures quality evaluation in situations where automation is impractical. Teachers can examine complete student responses, utilize grading tools, and filter submissions by class or question type using the Teacher Grading Dashboard. Teachers evaluate the student's response, issue a grade within the permitted range, and offer thorough commentary as part of the essay grading process. Crucially, when developing essay questions, OnTest gives teachers the ability to directly build unique rubrics, providing explicit grading standards that are shown during evaluation. These criteria aid in preserving uniformity and openness among various entries. The system immediately notifies students and updates the overall score after grading is finished.

#### Manual Grading

Grade essay and code submissions, provide feedback to students

The screenshot displays the 'Manual Grading' interface for an essay question. On the left, a 'Student Submissions' sidebar shows a list of students, with 'Gia Ngô' highlighted as 'Needs Grading'. The main area is titled 'Essay Grading' and includes a 'Save Grade' button. The 'Question' section contains a Vietnamese text prompt about 'Reconciliation' in React. The 'Student Answer' section shows the student's response, which is 124 words long. The 'Grading' section features a 'Manual Grading' tab, a score slider set to 5/10, and a feedback text area with a 'Good answer!' comment. A note at the bottom of the feedback area suggests providing specific feedback on strengths and areas for improvement.

**Student Submissions**  
(1 need grading)

Showing only ungraded submissions

**Gia Ngô** Needs Grading  
ngogia1581@gmail.com  
Submitted: May 30, 2025, 02:39 PM

**Gia Ngô**  
1 question(s) to grade • Submitted May 30, 2025, 02:39 PM

**Essay Grading**  
Grade the student's essay response and provide feedback

**Question**

"Trình bày chi tiết về khái niệm "Reconciliation" (Đối chiếu) trong React và vai trò của Virtual DOM trong quá trình này. Hãy giải thích cách cơ chế này giúp React tối ưu hóa hiệu suất cập nhật giao diện người dùng. Đồng thời, hãy thảo luận ít nhất hai (2) kỹ thuật phổ biến mà lập trình viên có thể áp dụng để tối ưu hóa hiệu suất render trong một ứng dụng React phức tạp, kèm theo ví dụ minh họa (nếu có thể)."

**Student Answer** 124 words

"Reconciliation" (đối chiếu, hòa giải) trong kế toán là quá trình so sánh và xác minh các số liệu hoặc dữ liệu từ hai nguồn khác nhau để đảm bảo chúng nhất quán và chính xác với nhau. Điều này giúp phát hiện những bất thường và sai sót, đảm bảo tính chính xác của số

**Grading**

Manual Grading

Score 5 / 10

Feedback

Good answer!

Provide specific feedback on strengths and areas for improvement

*Figure 4.3.2: UI for Manual Grading*

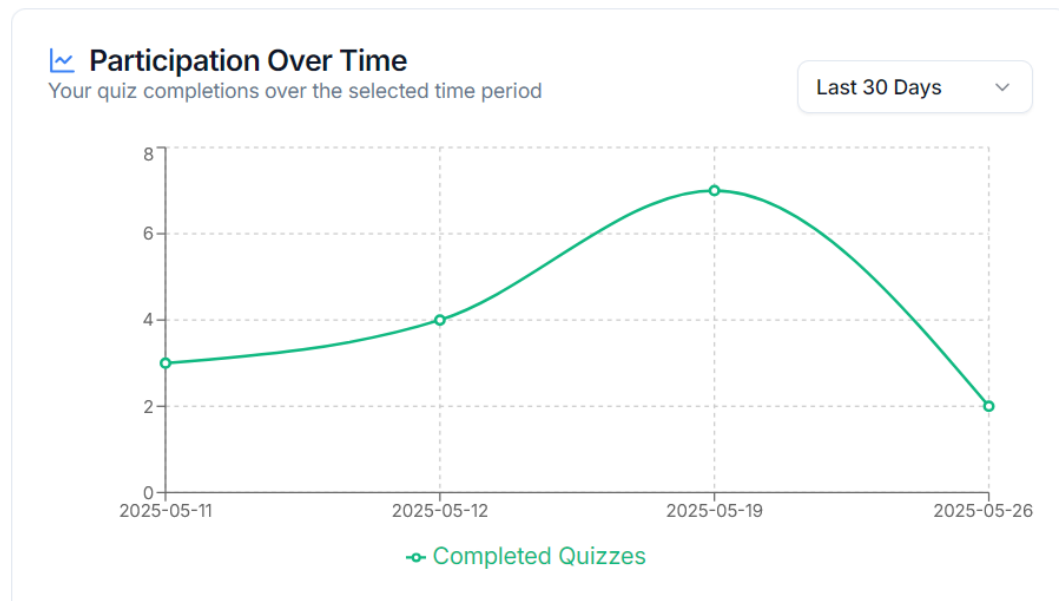
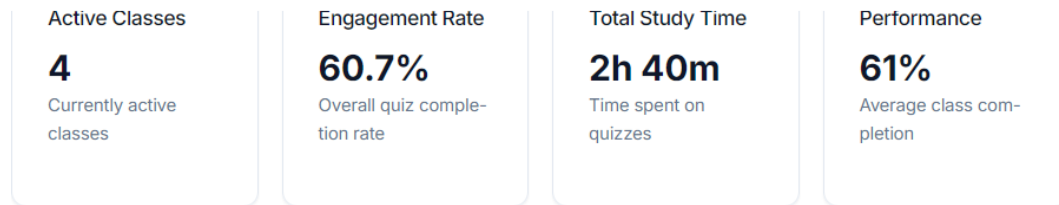
### 4.3.3 Score Calculation and Performance Analytics

A strong analytics and scoring system is used by OnTest to efficiently monitor student progress. The platform adds together all Answer.score values and compares them with the total number of potential points to determine the overall score once each question has been assessed. Based on the defined Quiz.passingScore, a pass/fail status is established, and QuizAttempt.score is changed appropriately.

#### Class Participation Analytics



Detailed analysis of your class participation and activities

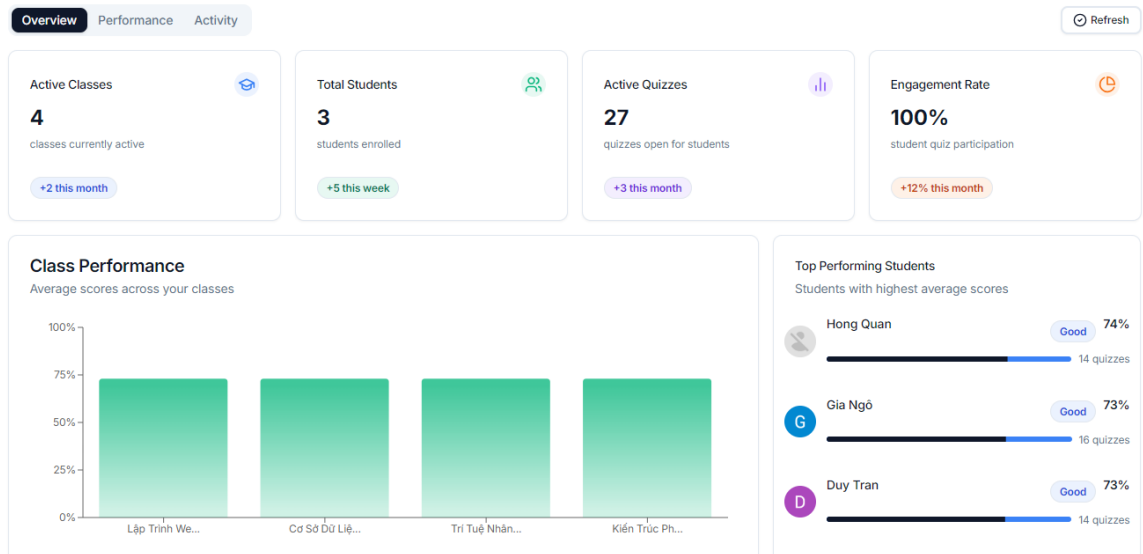


**Figure 4.3.3A: Class Participation and Performance Analytics**

The system also records performance data such the student's percentile rank in relation to peers, average response time per question, and total time spent (QuizAttempt.timeSpent) in addition to raw scores. Teachers may view individual and group progress using data-driven insights thanks to these insights, which are fed into the larger analytics modules, updating ScoreAnalytics per student, SubjectPerformance across subjects, and ClassPerformance for class-wide reporting.

## Teacher Dashboard

Overview of your classes, students, and educational performance metrics.



**Figure 4.3.3B: Comprehensive Analytics Dashboard – Teacher View**

## **CHAPTER V**

### **RESULTS**

#### **5.1 Results**

OnTest effectively integrated every essential feature needed for a cutting-edge online assessment system. The five main question categories supported by the system are essay, matching, multiple choice (MCQ), true/false, and short answer. There is a specific interaction interface and matching grading logic for each category. Robust algorithms are used to automatically evaluate multiple-choice questions and brief responses, supporting partial credit and adjustable penalties. Essay questions are manually assessed using a dedicated instructor interface that enables the construction of rubrics, the submission of scores, and thorough feedback.

The entire process of creating a quiz has been streamlined into four steps: creating questions, modifying settings, previewing the quiz, and entering basic information. Instructors can establish passing standards, time limits, and attempt restrictions. It is possible to link quizzes to particular classes and send students automated alerts. The class management system makes it possible to create both private and public classrooms, enroll students, offer quizzes, and make announcements.

The quiz-taking interface has been created to give students an easy-to-use and effective experience. It offers an easy-to-use navigation panel with progress monitoring and a countdown timer, as well as quizzes that are categorized (Upcoming, Available,

Completed, Expired). The system computes total scores, determines pass/fail status, and updates relevant data models upon input, whether human or automatic.

Additionally, thorough performance analytics have been incorporated. Percentile rank, average reaction time, and total time spent are among the metrics that are tracked and displayed. Both teachers and students have access to specific dashboards where they can examine subject-specific insights, class-wide outcomes, and score distribution, while students may keep an eye on their personal performance trends and growth.

Every feature listed above was successfully tested and put into use. To illustrate various implementations, sample interface screenshots and diagrams have been included in the preceding sections. The results verify that OnTest has achieved its main objectives, which were to provide a platform for assessments that is accurate, versatile, and easy to use.

## **5.2 Limitations and Future Enhancements**

In general, OnTest has accomplished the majority of its original objectives, such as providing a useful quiz system, class administration features, automatic grading for structured questions, and performance analyses. Concrete implementations and previously displayed interface designs support these findings. Nevertheless, there are still certain restrictions, like manual essay grading, simple short answer assessment, inadequate test security, restricted LMS integration, and poor mobile coverage. To increase system scalability and dependability, these gaps point to important areas that need to be improved in the future.

## **CHAPTER VI**

### **CONCLUSION**

#### **6.1 Conclusion**

OnTest, an online assessment platform that allows teachers to create and administer quizzes, automate grading for structured question types (like multiple choice, true/false, matching, and short answers), and analyze student performance through interactive dashboards, was successfully designed and implemented by this project. Students can efficiently complete quizzes with the system's user-friendly interface and support for class-based assignment management.

From a research and development standpoint, OnTest offers a workable solution that combines data-driven insights, real-time feedback, and contemporary assessment methods. It demonstrates how organized assessment routines may be improved and digitalized with little manual involvement, all the while maintaining flexibility in essay grading and qualitative comments. The platform is a pertinent resource for upcoming scholarly work in formative assessment, intelligent tutoring, and e-learning systems due to its extensible design and modular grading logic.

#### **6.2 Recommendations**

Future improvements should address existing issues including the lack of automatic essay scoring, the restricted semantic assessment in short responses, and the scant test security measures to improve OnTest' resilience and scalability. Enhancing the

platform's proctoring, tab-switch detection, and biometric login capabilities would make it more appropriate for high-stakes testing situations. Furthermore, enhancing the mobile experience and increasing integration compatibility with external LMSs (like Moodle and Canvas) would increase its usefulness in actual educational contexts.



## LIST OF REFERENCES

- [1] “Testportal: Skills and knowledge assessment platform,” Testportal.  
<https://www.testportal.net/> (accessed May 29, 2025).
- [2] ClassMarker, “Online testing & quiz maker features,” ClassMarker.  
<https://www.classmarker.com/online-testing/quiz-features> (accessed May 29, 2025).
- [3] “Secure Online exam Platform | Scalable & Customisable – SyNAP.”  
<https://synap.ac/solutions/online-exam-platform/> (accessed May 29, 2025).
- [4] S. Stoyanova and V. Giannouli, “Online Testing as a Means of Enhancing Students’ Academic Motivation during the Coronavirus Pandemic,” *Education Sciences*, vol. 13, no. 1, p. 25, Dec. 2022, doi: 10.3390/educsci13010025.
- [5] Admin, “Effectiveness of Personalized Learning: Statistics on outcomes in diverse educational settings,” *Matsh People Development*, Dec. 28, 2024.  
<https://www.matsh.co/en/statistics-on-personalized-learning-effectiveness/>.
- [6] KnowledgeWorks, “Personalized learning and technology: 4 things to keep in mind - KnowledgeWorks,” KnowledgeWorks, Apr. 03, 2025.  
<https://knowledgeworks.org/resources/personalized-learning-role-technology/> (accessed May 29, 2025).
- [7] Digiexam, “Online assessment tool in education,” Digiexam.  
<https://www.digiexam.com/online-assessment-tool-in-education/> (accessed: May 29, 2025).

- [8] Speed Exam, “Flexible online assessment platform for learning - SpeedExam,” Speed Exam. <https://www.speedexam.net/blog/online-assessment-platform/> (accessed May 29, 2025).
- [9] L. Romano, “Top assessment tools for K-12 education,” Atomic Blog, Jan. 13, 2025. <https://blog.atomicjolt.com/top-assessment-tools-for-k-12-education/> (accessed May 29, 2025).
- [10] H. Chen and P. a. S. Ward, “Predicting student performance using data from an auto-grading system,” arXiv.org, Feb. 2021, doi: 10.5555/3370272.3370297.
- [11] G. O’Neill and L. Padden, “Diversifying assessment methods: Barriers, benefits and enablers,” Innovations in Education and Teaching International, vol. 59, no. 4, pp. 398–409, Feb. 2021, doi: 10.1080/14703297.2021.1880462.
- [12] AnalytikUS, “The future of Education and AI: Automated assessment and feedback,” analytikUS, Jun. 02, 2024. <https://www.analytikus.com/post/the-future-of-education-and-ai-automated-assessment-and-feedback>.
- [13] E. Balla, “Automated Grading Systems: How AI is Revolutionizing Exam Evaluation,” Data Science Central, Nov. 30, 2024. <https://www.datasciencecentral.com/automated-grading-systems-how-ai-is-revolutionizing-exam-evaluation/> (accessed May 30, 2025).
- [14] A. P. Global, “Virtual classroom management 101,” ASU Prep Global, May 01, 2025. <https://www.asuprepglobal.org/news/virtual-classroom-management/> (accessed May 30, 2025).

- [15] J. Han, K. H. Kim, W. Rhee, and Y. H. Cho, “Learning analytics dashboards for adaptive support in face-to-face collaborative argumentation,” *Computers & Education*, vol. 163, p. 104041, Oct. 2020, doi: 10.1016/j.compedu.2020.104041.
- [16] S. Leonard, “Automated Grading for subjective Assessments: Challenges and solutions,” TAO, Feb. 12, 2025. <https://www.taotesting.com/blog/automated-grading-for-subjective-assessments-challenges-and-solutions/> (accessed May 30, 2025).
- [17] P. Vaid and P. Vaid, “10 Key challenges in LMS Management - Mastering LMS,” MapleLMS, Jun. 14, 2024. <https://www.maplelms.com/blog/10-key-challenges-in-lms-management-mastering-lms/> (accessed May 30, 2025).
- [18] B. De, “Traditional learning vs. online learning,” eLearning Industry, Mar. 11, 2025. <https://elearningindustry.com/traditional-learning-vs-online-learning> (accessed May 30, 2025).
- [19] “Online assessment tools: 5 major benefits.” <https://www.educationadvanced.com/blog/online-assessment-tools-5-major-benefits> (accessed May 30, 2025).
- [20] S. Leonard, “4 Common challenges when shifting to digital assessments—and how to overcome them,” TAO, Jul. 23, 2024. <https://www.taotesting.com/blog/4-common-digital-assessment-challenges-and-their-solutions/> (accessed May 30, 2025).
- [21] JD\_Webdev, “Exploring the challenges and opportunities of online assessments - the E-Assessment Association,” The e-Assessment Association, Jul. 18, 2023. <https://www.e->

[assessment.com/news/exploring-the-challenges-and-opportunities-of-online-assessments/](https://assessment.com/news/exploring-the-challenges-and-opportunities-of-online-assessments/)

(accessed May 30, 2025).

[22] A. L. K. P. Rheubert and Justin L., “Redefining Assessment: Empowering Students through a Blended Approach,” Faculty Focus | Higher Ed Teaching & Learning, Nov. 04, 2024. [https://www.facultyfocus.com/articles/educational-assessment/redefining-](https://www.facultyfocus.com/articles/educational-assessment/redefining-assessment-empowering-students-through-a-blended-approach/)

[assessment-empowering-students-through-a-blended-approach/](https://www.facultyfocus.com/articles/educational-assessment/redefining-assessment-empowering-students-through-a-blended-approach/) (accessed May 30, 2025).

[23] K. Glackin, “What student assessment looks like in a K-12 blended environment,” PowerSchool, Mar. 20, 2025. [https://www.powerschool.com/blog/what-student-](https://www.powerschool.com/blog/what-student-assessment-looks-like-in-a-k-12-blended-environment/)  
[assessment-looks-like-in-a-k-12-blended-environment/](https://www.powerschool.com/blog/what-student-assessment-looks-like-in-a-k-12-blended-environment/) (accessed May 30, 2025).

[24] C. Lee, “Effective test questions: What instructors need to know,” Turnitin, Aug. 11, 2023. [https://www.turnitin.com/blog/what-makes-effective-test-questions-and-answers-](https://www.turnitin.com/blog/what-makes-effective-test-questions-and-answers-for-assessments-what-instructors-and-administrators-need-to-know)  
[for-assessments-what-instructors-and-administrators-need-to-know](https://www.turnitin.com/blog/what-makes-effective-test-questions-and-answers-for-assessments-what-instructors-and-administrators-need-to-know) (accessed May 30, 2025).

[25] S. Wijayanti and H. Prasadja, “Quizizz and Kahoot! for Online-Learning Engagement in Elementary School,” Proceedings of the 6th Batusangkar International Conference, BIC 2021, 11 - 12 October, 2021, Batusangkar-West Sumatra, Indonesia, Jan. 2022, doi: 10.4108/eai.11-10-2021.2319534.

[26] B. R. Gryshuk, “Moodle vs Canvas: LMS Comparison for Educators 2025.” <https://www.educate-me.co/blog/moodle-vs-canvas> (accessed May 30, 2025).

- [27] Dataintel, R. Sharma, and Dataintel, “Online Assessment Software Market Report | Global Forecast from 2025 to 2033,” Dataintel, Apr. 05, 2024. <https://dataintel.com/report/global-online-assessment-software-market>
- [28] Reshaping Learning with Next Generation Educational Technologies. 2024. doi: 10.4018/979-8-3693-1310-7.
- [29] L. Naidoo, “AI vs. Human: Exam grading accuracy comparison,” GRAiDY, Mar. 12, 2025. <https://www.graidy.tech/exam-grading-accuracy-comparison>
- [30] X. Wu, P. P. Saraf, G.-G. Lee, E. Latif, N. Liu, and X. Zhai, “Unveiling Scoring Processes: Dissecting the Differences between LLMs and Human Graders in Automatic Scoring,” arXiv (Cornell University), Jul. 2024, doi: 10.48550/arxiv.2407.18328.
- [31] U.-I. Technology, “Auto Grading with Regular Expressions,” Instructor Help | Instructional Technology Help Center, Sep. 27, 2024. <https://help.intech.arizona.edu/article/511-auto-grading-with-regular-expressions> (accessed May 30, 2025).
- [32] F. F. L. M. Rosmansyah Atina Putri, Dana Waskita, Tri Sulistyaningtyas, Arry Akhmad Arman, Yusep, “Automated Short-Answer Grading using Semantic Similarity based on Word Embedding,” IJTech - International Journal of Technology. <https://ijtech.eng.ui.ac.id/article/view/4651> (accessed May 30, 2025).
- [33] S. Lee, “5 Key Metrics: Classroom Analytics Driving Education Success.” <https://www.numberanalytics.com/blog/5-key-metrics-classroom-analytics-driving-education-success> (accessed May 30, 2025).

[34] H. Peter, “Enhancing Student Learning Outcomes with Data Analytics: A Guide for Educators,” Hyperstack, Feb. 12, 2025. <https://thehyperstack.com/blog/using-data-analytics-enhance-student-learning-outcomes>.