EXPLORSAP - AN ONLINE LEARNING PLATFORM

HUANG JINGXI, Kamsuriah Ahmad

Faculty of Information Science & Technology

Universiti Kebangsaan Malaysia

43600 Bangi, Selangor

Abstract

ExplorSap ialah platform pembelajaran yang menyepadukan kuiz pendidikan dengan Sistem Maklumat Pengurusan (MIS) untuk meningkatkan penglibatan pengguna dan hasil pembelajaran. Platform ini menangani isu seperti penglibatan yang rendah, kekurangan pemperibadian dan maklum balas yang tidak mencukupi dalam pendidikan dalam talian tradisional dengan menyediakan papan pendahulu masa nyata, cabaran penyesuaian dan laluan pembelajaran yang diperibadikan. Dibangunkan menggunakan JavaScript dan HTML5 untuk modul permainan dan HTML5 untuk modul web, ExplorSap menawarkan kuiz dinamik dengan elemen interaktif, antara muka web mesra pengguna untuk menjejak kemajuan dan alatan pentadbiran untuk mengurus kandungan dan peranan. Pembangunan mengikut metodologi Agile, membolehkan pelaksanaan berulang dan maklum balas berpusatkan pengguna. Fungsi utama termasuk penyertaan kuiz dinamik dengan maklum balas segera, pengesyoran pembelajaran diperibadikan dipacu MIS, sistem ganjaran dengan mata dan lencana yang boleh ditebus serta alatan guru dan pentadbir yang komprehensif untuk penciptaan dan pengurusan kandungan. Setelah selesai, ExplorSap menyasarkan untuk menyediakan platform interaktif berskala yang memupuk motivasi, menyokong pendidikan yang disesuaikan dan memanfaatkan cerapan terdorong data untuk meningkatkan kecekapan

pembelajaran.

Abstract

ExplorSap is an learning platform that integrates educational quizzes with a Management Information System (MIS) to enhance user engagement and learning outcomes. The platform addresses issues such as low engagement, lack of personalization, and insufficient feedback in traditional online education by providing real-time leaderboards, adaptive challenges, and personalized learning paths. Developed using JavaScript and HTML5 for the game module and HTML5 for the web module, ExplorSap offers dynamic quizzes with interactive elements, user-friendly web interfaces for tracking progress, and administrative tools for managing content and roles. The development follows an Agile methodology, enabling iterative implementation and user-centered feedback. Key functionalities include dynamic quiz participation with instant feedback, MIS-driven personalized learning recommendations, a reward system with redeemable points and badges, and comprehensive teacher and admin tools for content creation and management. Upon completion, ExplorSap aims to provide a scalable, interactive platform that fosters motivation, supports tailored education, and leverages data-driven insights to improve learning efficiency.

1.0 INTRODUCTION

ExplorSap is an educational platform that combines management information systems (MIS) and simulate game interactive animation mechanism to provide a personalized, interactive learning experience.In the context of ExplorSap, the integration of a Management Information System (MIS) is essential for capturing and analyzing user behavior, providing personalized content recommendations, and enhancing operational efficiencies (O'Brien & Marakas 2011).The Management Information System (MIS) module in ExplorSap is used for user data management and learning progress tracking, similar to the Student Information System. Through MIS, the platform can effectively record and manage users' personal information, learning data, answer records and each

user's personalized learning path.MIS integration in ExplorSap enhances the platform's capacity to provide real-time insights into user performance, improving personalized feedback and overall user engagement (Laudon & Laudon 2018).Data recording and storage: MIS module not only collects and stores the user's answer history, learning time, knowledge point and other data in real time, but also stores the user's game achievements, competition rankings and other competitive data, to provides the basis for subsequent analysis and content recommendation.

1.1 PROBLEM STATEMENT

Traditional quizzes often fail to sustain user interest due to their limited interactivity and engagement. Without a data analytics system to analyze user data and personalize it, users quickly lose interest.Moreover, in most cases, positive and negative emotions during learning could promote and hinder learning performance, respectively (Forster et al. 2015; Loderer et al. 2019).The overall efficiency of the learning process decreases, and retention rates are low.

Without an integrated management information system (MIS), these traditional platforms also lack the ability to systematically collect, manage, and analyze user data. The lack of intuitive auxiliary data to guide users' learning leads to their inability to find and improve their learning shortcomings, and over time, users will reduce their dependence on the platform because their learning ability has not been improved.

In addition, the lack of data management further affects the efficiency of the learning process, as users do not have the support they need to improve on weaknesses. The lack of customized experience will reduce the effectiveness of the platform, making it difficult for users to retain information for a long time, which will reduce the educational character of the platform, and finally lead to the imbalance of the functional function of the platform in education and competition.

1.2 OBJECTIVE

At this stage,ExplorSap project is to design and develop an educational platform that combines simulation gamification elements with a management information system (MIS) to provide a personalised learning experience with data collection through strategy-based quiz games,the target users are students aged 12-15 years old.

The following is the details about SMART:

i.S – Specific

The focus of this project is to develop a test platform combined with simulate game interactive animation mechanism, it will has an management information system (MIS) as the back-end system for data support. The main task of MIS is to manage user data, track user behavior and provide personalized learning feedback. By collecting users' performance data and identifying knowledge weaknesses, the system provides customized content to fetch each user's learning needs.

ii.M – Measurable

MIS will evaluate the effectiveness of the platform in promoting learning outcomes by collecting user data on promotion and engagement behavior patterns. Key indicators in the assessment include the following:

Learning progress tracking: The system will record the user's improvement in learning performance over time, with special attention to previously identified areas of weak knowledge. The system will evaluate the effectiveness of personalized content recommendations by improving accuracy, reducing completion time, and successfully completing targeted learning modules.

Engagement Metrics: To gain insight into and increase user engagement, MIS will track the number of active users, the rate of quiz completion, and the frequency of user engagement in contest interactions (such as the number of leaderboard ranking queries).

Knowledge retention analysis: The platform will periodically evaluate the knowledge retention of users and compare their scores after initial learning with those after repeated exposure to similar content.

iii.A – Achievable

The project schedule will span two semesters. The initial design phase will be completed by the end of the first semester and will include a comprehensive project outline, requirements analysis, and technical specification documentation. The second phase, from mid-March to mid-July, involves the full implementation of core functions, including quiz modules, user data management, and personalized learning feedback mechanisms,, and the development will use such as node.js,HTML5 (H5),CSS,JavaScript to ensure compatibility and feasibility.

iv.R – Relevant

ExplorSap provides a challenging and flexible learning atmosphere. Utilizing the growth in digital education, the need for interactive customized content becomes even more crucial. ExplorSap combines game-based analysis with personalized learning pathways to support users in a challenging global learning environment.

v.T – Duration (Timed)

Design and development are carried out over the site's two-week stages. The design period may end this quarter in January 2025 and includes project planning, literature review, requirements analysis, and detailed standards documentation.

The growth phase will commence in the middle of March 2025 and extend until the end of July. It will rely on feature implementation, testing, and efficiency. Standard progress assessments and milestone checks are conducted to ensure that each task is completed on time. Following this period, the platform is ultimately deployed.

2.0 LITERATURE REVIEW

Gamification in education refers to the incorporation of game elements such as points, badges, and leaderboards into learning environments, aiming to increase student engagement and motivation. As highlighted by Caponetto et al. (2014), gamification has shown positive impacts in fostering collaboration, self-regulated learning, and creativity. However, its effectiveness is often limited when used in isolation without the support of data-driven customization. Laudon and Laudon (2018) emphasize that Management Information Systems (MIS) can complement gamified environments by

enabling structured data collection, real-time tracking, and personalized content delivery. MIS allows platforms to dynamically adapt to users' needs through data analytics, thus addressing individual learning gaps more effectively than static gamified systems.

ExplorSap was designed in response to this integration challenge. The system leverages MIS to record and analyze user behaviors in real time, enabling adaptive quiz delivery and performance feedback. Unlike traditional gamified platforms that rely solely on superficial engagement, ExplorSap uses MIS to generate personalized learning paths, ensuring users receive targeted content based on their ongoing performance.

With the rise of digital education, studies such as those by Halverson and Graham (2019) and Ma et al. (2018) suggest that measuring student engagement in online environments requires both qualitative and quantitative indicators. These include not only user satisfaction or feedback but also behavioral data such as time-on-task, response accuracy, and participation trends—areas that can be effectively tracked through MIS. ExplorSap incorporates these metrics, enhancing its ability to respond dynamically to learner performance and engagement patterns.

Previous systems such as Kahoot and Duolingo have demonstrated the motivational potential of gamification, yet they fall short in delivering personalized feedback or adapting to diverse learner profiles. MIS-based platforms like Edmodo or Google Classroom offer administrative structure but often lack real-time learning interaction and adaptive feedback mechanisms. ExplorSap addresses these limitations by combining the motivational features of gamification with the structural strengths of MIS, supporting real-time contests, adaptive challenges, and detailed performance analytics.

Furthermore, challenges such as scalability, lack of real-time feedback, and rigid gamified structures have been noted across prior studies. ExplorSap responds by

adopting a distributed MIS architecture that ensures scalability while preserving responsiveness and individualization. It also includes storyline-based design elements inspired by narrative learning research (Wang & Lieberoth, 2016), further enhancing immersion and user retention.

In summary, ExplorSap builds upon the insights of earlier educational technologies by fusing gamification with MIS to form a responsive, scalable, and personalized digital learning environment. It moves beyond the limitations of static quiz tools by leveraging data to provide targeted content, monitor engagement, and improve the overall learning experience in real time.

3.0 METHODOLOGY

The ExplorSap project will use agile development model ,it will divides projects into iterative cycles to ensure flexibility, adaptability, and continuous improvement throughout the development process. Each cycles will focus on implementing specific features, conducting thorough testing, and integrating user feedback.

3.1 System model

This use case diagram outlines the core functionalities and role-based interactions in the ExplorSap system. Students can register, join quizzes, view leaderboards, earn badges, and access personalized learning paths. Teachers register and log in, after which the system verifies their identity before allowing them to manage subjects and quizzes and perform grading. Admins oversee system content and role management. The diagram also shows how certain features extend or include others, reflecting functional dependencies and user flow logic.

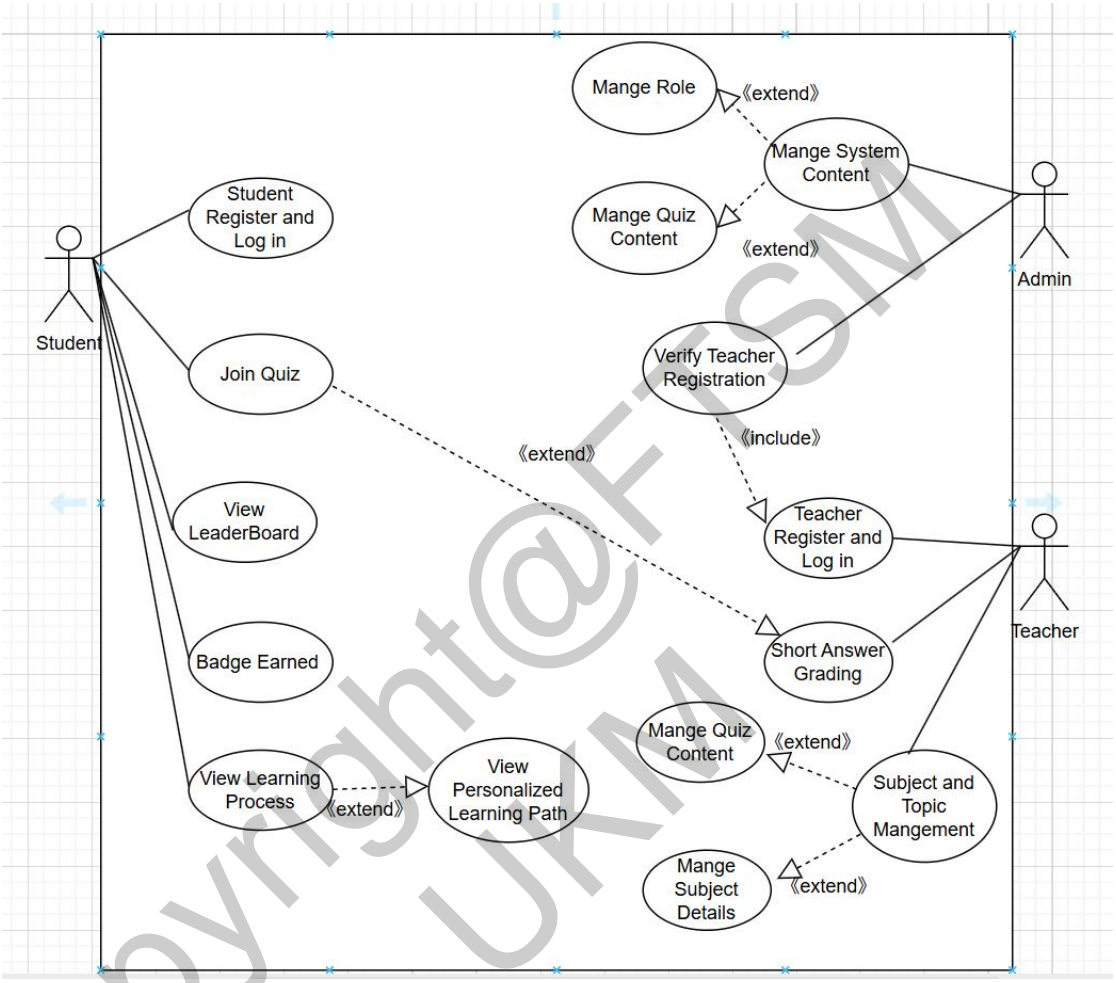

Figure 3.1 shows Use Case Diagram for ExplorSap System

3.2 ARCHITECTURAL DESIGN

The figure shows a layered architecture of the web system, including the frontend (static pages, JavaScript, CSS), backend (Express server, APIs, authentication), and database layer (MySQL, file storage).

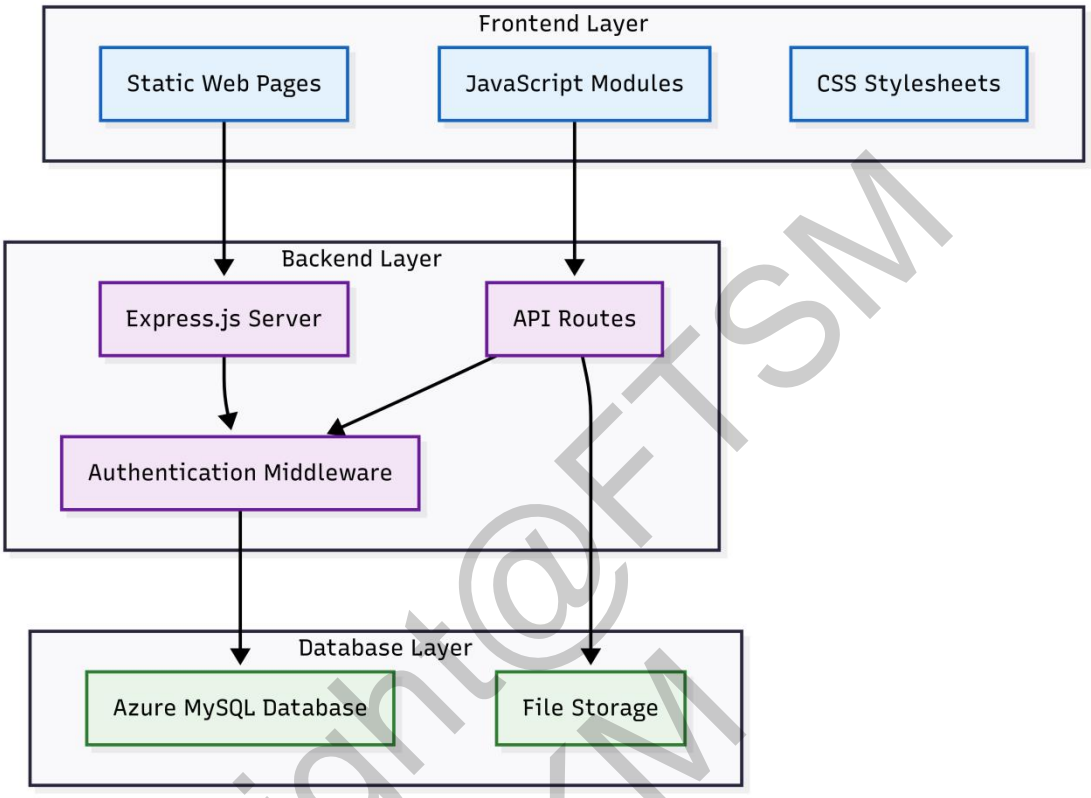

Figure 3.4 shows System Architecture Diagram

3.3 System Architecture Design

The system adopts Model-View-Controller (MVC) architecture, with the goal of achieving clear logic, module separation, and easy development, testing, and maintenance. Through the MVC architecture, the functional independence of the student test module and other modules can be ensured, while maintaining the overall coordination of the system.

i.Model layer (Model):

Function: Process business logic and data management,Store user information of students and teachers,Manage test data, including questions, answers, and test scores, Process student points and reward logic.

Core module:User data management,Test data storage,Reward distribution rules.

ii.View layer (View):

Function: Provide user interface and interactive experience.

Display student test interface, including questions, timers, and real-time feedback, Teacher interface for creating and publishing tests,Display of real-time leaderboards and test completion reports.

Core module:Test interface design,Leaderboard display module.

iii.Controller layer (Controller):

Function: Serves as a bridge between model and view,Process student test participation and answer submission requests,Call the model layer for data processing and return the results to the view layer.

Core modules:Student test process control,Test result processing.

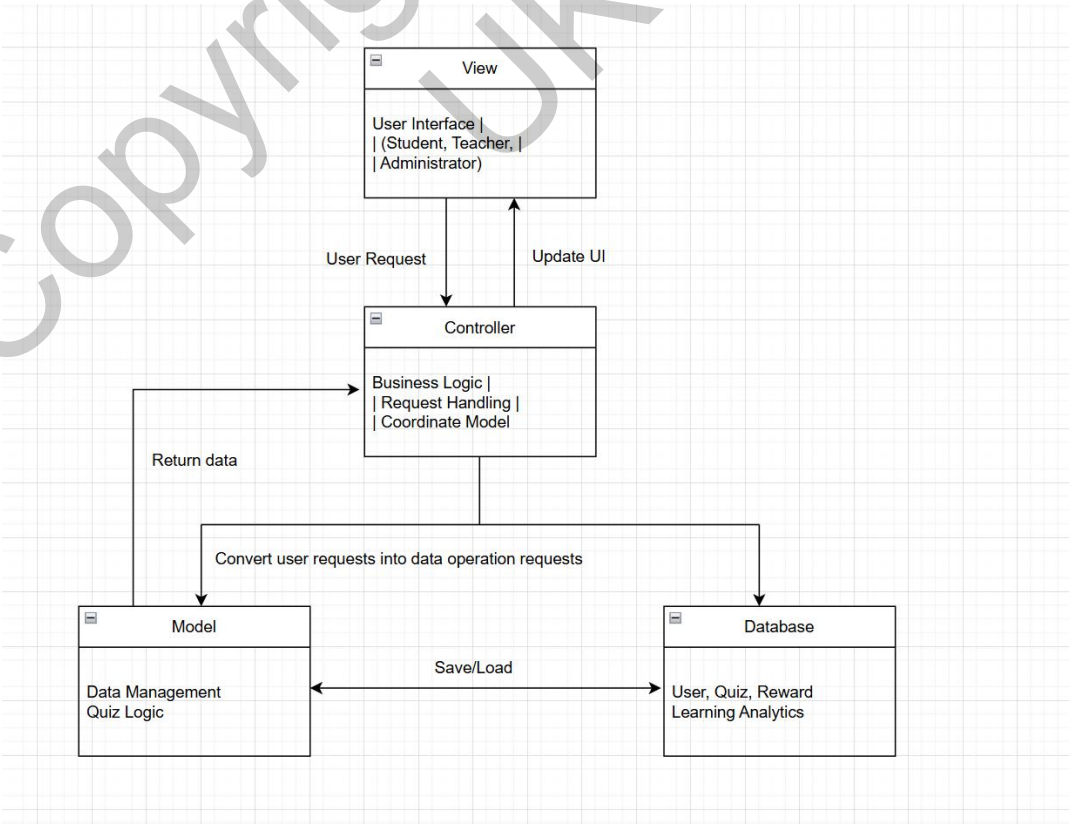

Figure 3.5 shows Model-View-Controller (MVC) type design architecture

3.4 DATABASE DESIGN

To ensure efficient and reliable data storage and management of the ExplorSap system, the database design adopts an object-oriented approach, using class diagrams and data dictionaries to clearly describe the system's core data structure and its relationships. The goals of this design are:Define the attributes of all core entities in the system (such as users, quizzes, rewards, etc.) and their relationships;Provide a clear data layer architecture to provide a consistent reference for system development and maintenance.

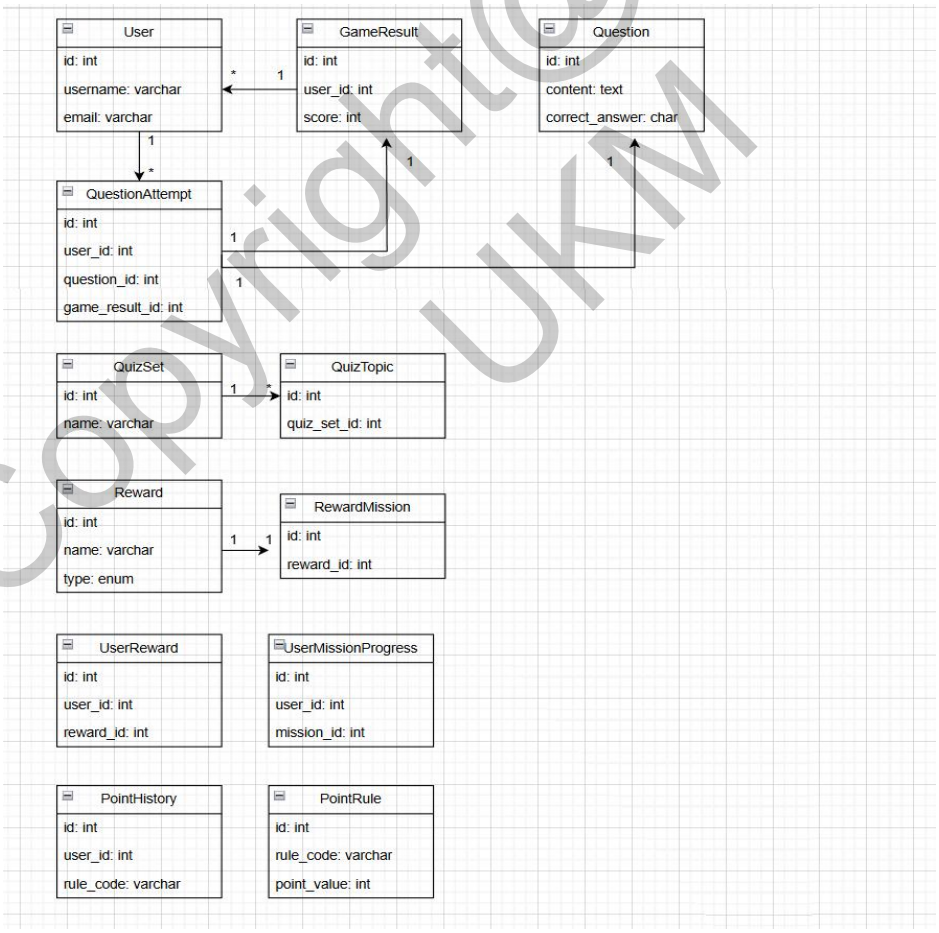

Figure 3.6 shows Class Diagram design of the ExplorSap System

4.0 RESULTS

4.1 Overview of Development Approach

The ExplorSap system was developed using a modular full-stack JavaScript architecture, following a front-end-back-end separation approach. The front-end is built with static HTML pages enhanced by Bootstrap for styling and jQuery for dynamic interactions. The back-end is built using Node.js + Express, while MySQL is used for persistent data storage.

The key development steps included:

i. Database Design and Setup:Tables were created to manage users, quiz questions, attempts, learning progress, and leader board data. The back-end connects to the database using mysql2 with promise-based queries.

ii. Back-end API Development :RESTful API endpoints were implemented using Express to support user registration/login, quiz interaction, result submission, and leaderboard data retrieval. Authentication was handled with JWT, and bcrypt was used for secure password hashing.

iii. Front-end Page Implementation:Pages like login.html, admin.html, learning-progress.html, and leaderboard.html were built with HTML, styled with Bootstrap, and dynamically connected to APIs using jQuery's AJAX and fetch functions.

iv. Testing and Debugging:Functionality and API responses were tested using Postman to ensure correct input/output formats, and error handling logic was added across routes to catch and log issues during development.

v. Security Considerations:JWTs were used for secure session handling, ensuring that only authenticated users could access certain routes. Passwords were hashed with bcrypt before being stored in the database.

4.2 Critical Modules and Code Segments

HTML Pages (teacher.html, class-manage.html, profile.html, etc.)

Provide the main user interface for teachers, students, and administrators.Includes dashboards, class management, subject management, quiz management, and user profiles etc.

i. Student Dashboard Interface

Figure 4.1 shows the Dashboard interface after the user logs in, which provides a course search bar and an overview of the learning modules. Each course is displayed in the

form of a card, with a course description and a completion progress bar. The top navigation bar has a tab bar switching function.

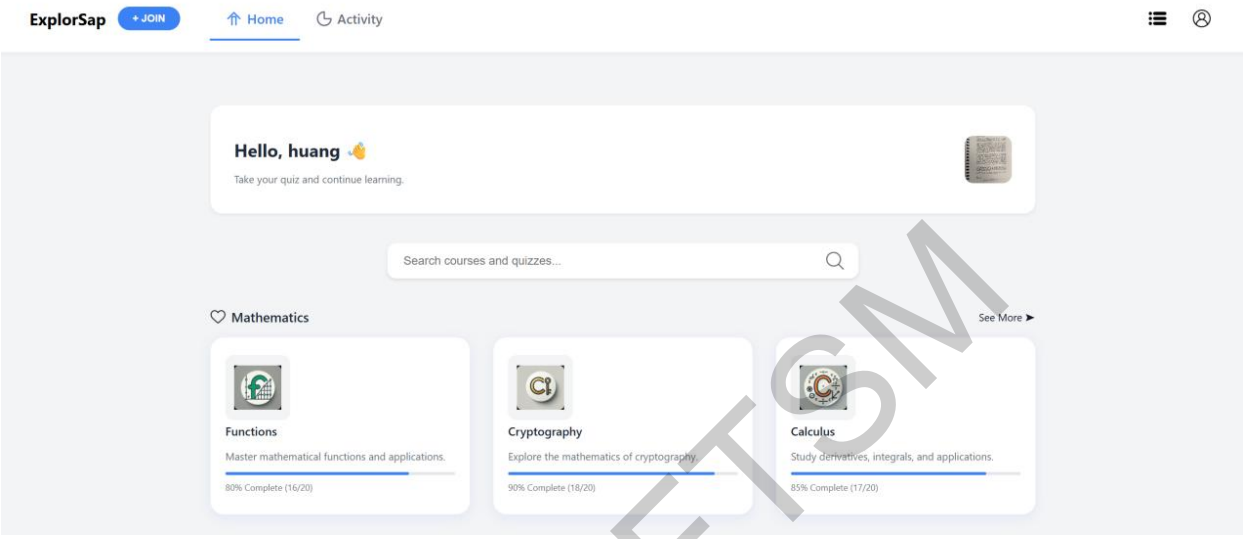

Figure 4.1 Student's Dashboard Interface

Figure 4.2 shows the tab bar on the Dashboard page switches to recent activity, which displays the user's recent test activity records. Each card shows the course name, completion time, and progress percentage, and the recommendation area below shows the best score and personalized learning suggestions.

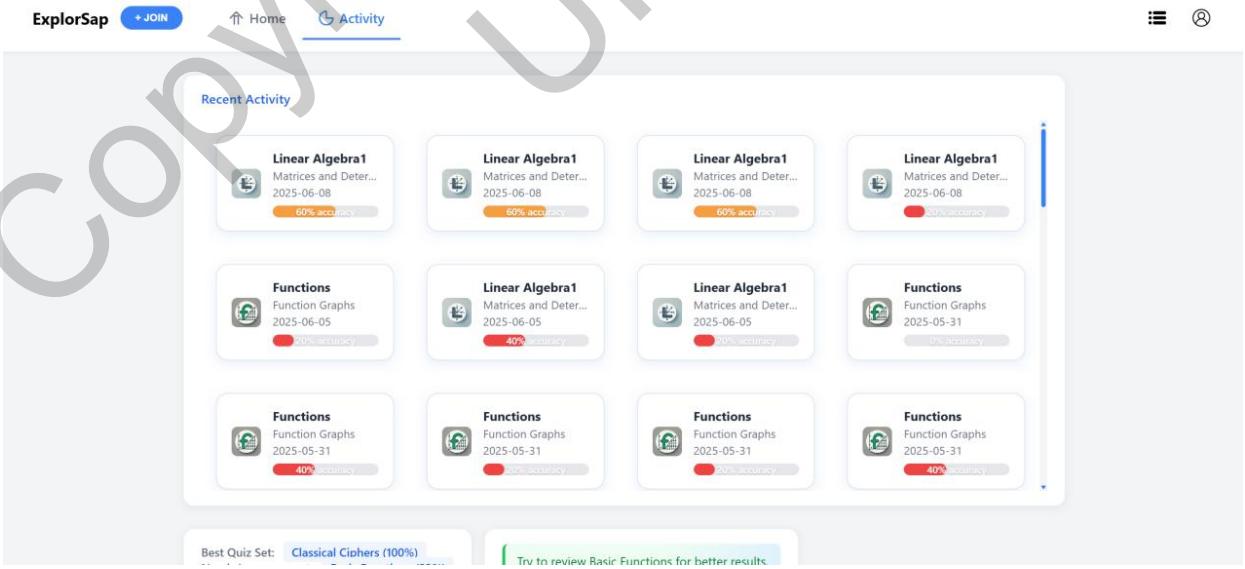

Figure 4.2 Student's Dashboard Interface

ii. Teacher page Interface

Figure 4.3 shows teacher Interface.The interface displays the student information of each class by class number (such as 1-1, 1-2). Each student has a card with his/her name and a graphical statistical bar of his/her learning performance.

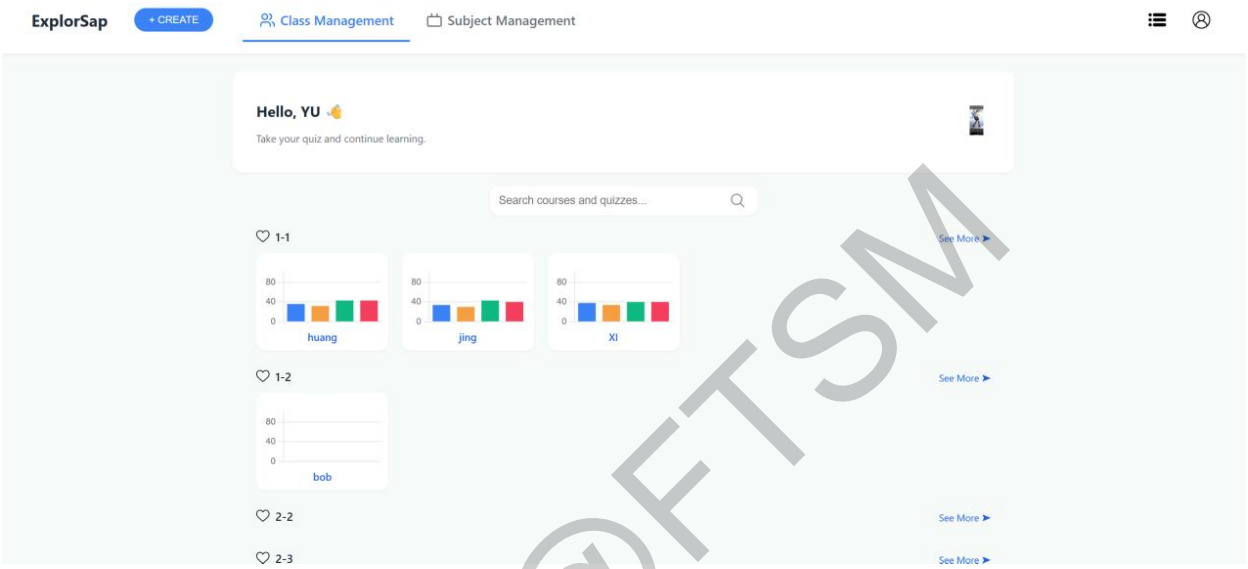

Figure 4.3 Teacher Interface

Figure 4.4 shows the subject management section of the teacher interface, which displays the basic information of each course in a table format, including fields such as name, description, category, cover image path, status, etc. Teachers can use the operation bar to change the course publishing status, save changes, delete courses, or view details.

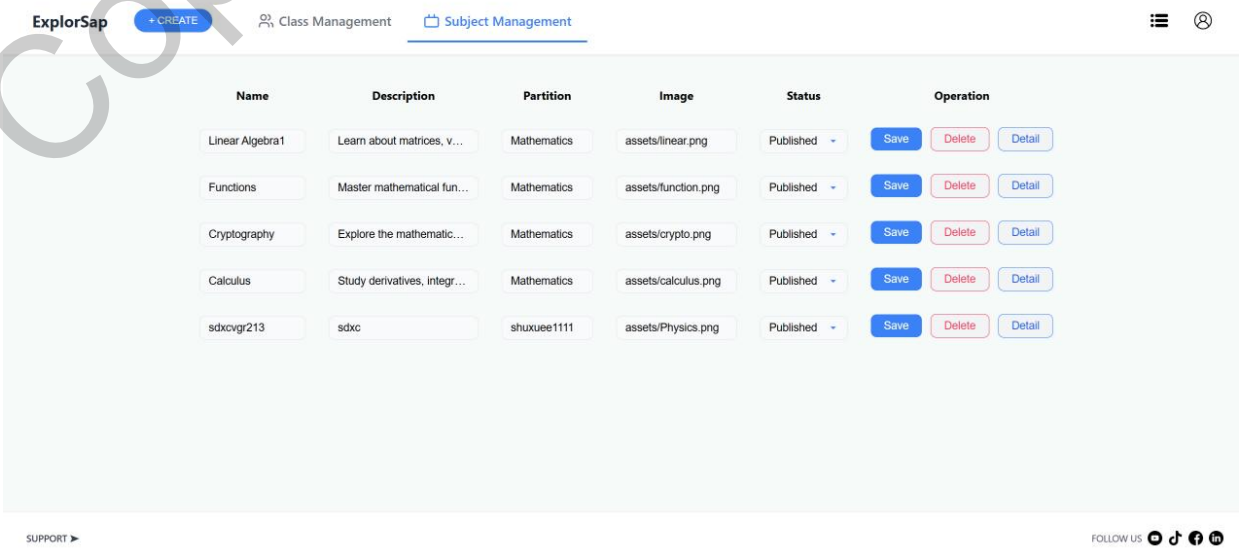

Figure 4.4 Teacher Interface

iii. Admin panel Interface

The developer creates a super administrator when the system is initialized. When the system is initially installed or deployed, the system automatically or manually sets a super administrator account.

Usually, it only needs to be manually inserted into the database or set through the installation process. The super administrator assigns other administrators. After the super administrator logs in to the back-end, he can assign administrator rights to certain user accounts through the "User Management" module.

Some systems allow users to apply to become administrators, and the back-end administrator approves and authorizes them.

Figure 4.5 shows the user management section of the Admin Interface, which supports filtering the user list by user name, role, and date. Administrators can perform operations such as batch deletion and batch password reset for users, and provide separate "Reset Password" and "Delete" buttons to facilitate individual account management.

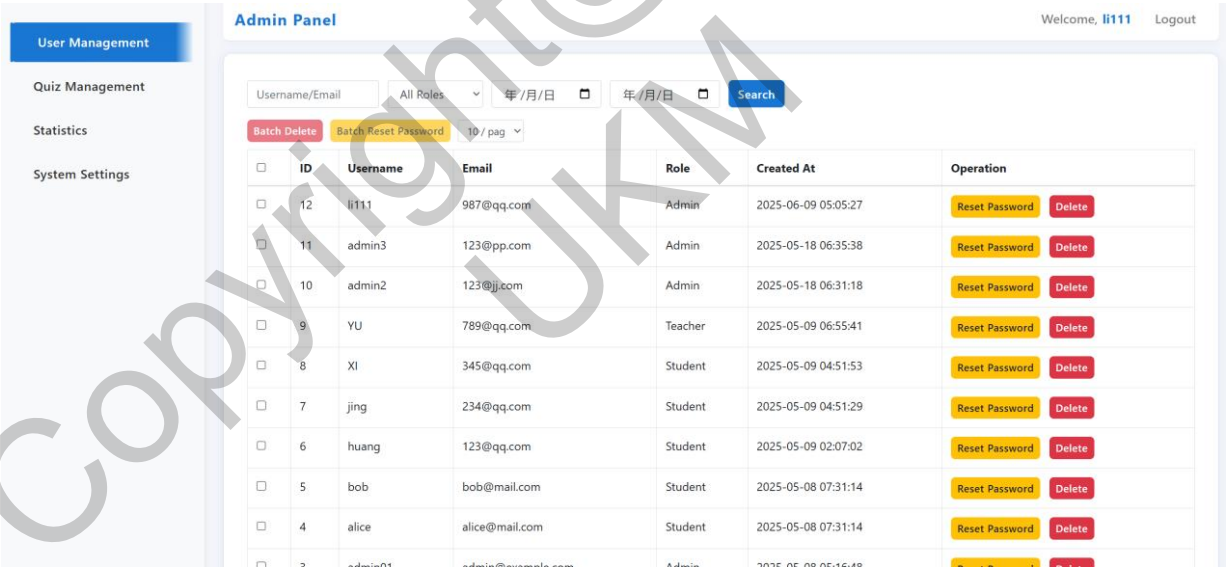

Figure 4.5 User Management in Admin Interface

Figure 4.6 shows the quiz management section of the Admin Interface, which supports viewing the content of questions by subject and topic, and can add, edit, and delete each question. It also provides difficulty screening and search functions.

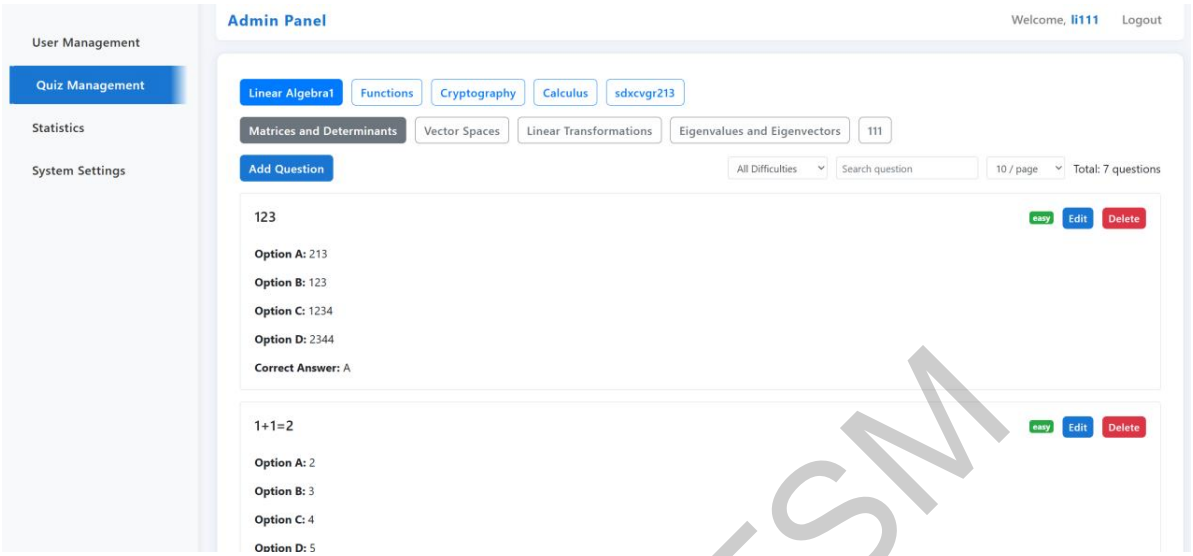

Figure 4.6 Quiz Management in Admin interface

Figure 4.7 shows the Statistics section of the Admin Interface, which displays the statistical analysis of users' answering behavior in the system, including the accuracy rate and average time of answering questions, active user trend chart, answering heat map and popular subject ranking.

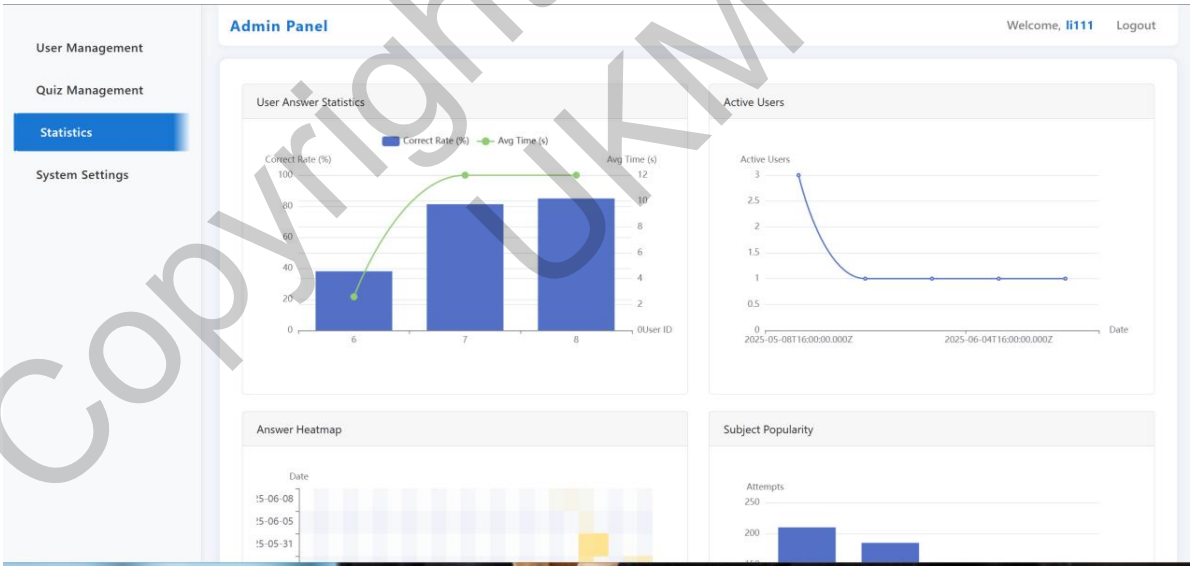

Figure 4.7 Statics in Admin Interface

iv. Animation interaction of Quiz interface

Figure 4.8 shows the racing animation interactive effect of the Quiz Interface, the sprint effect for correct answers and the dynamic prompt for incorrect answers, the dynamic progress bar rendering at the bottom, and the celebration animation after completion.

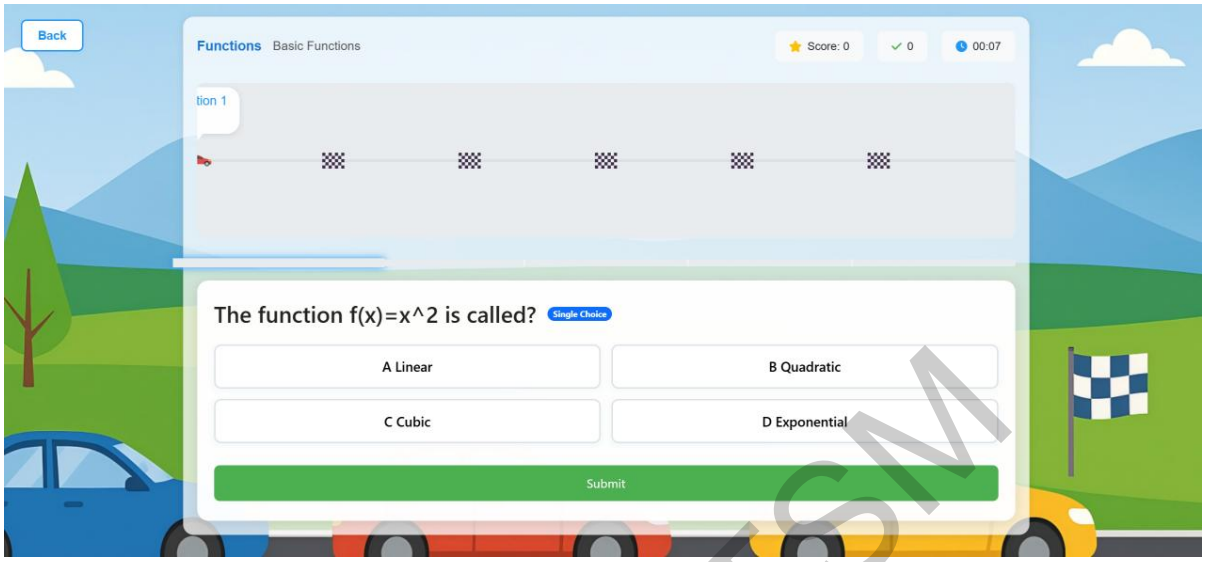

Figure 4.8 Quiz Interface

v. Learning process interface

Figure 4.10 shows the learning process interface. The line graph shows the trend of users' answer scores over a period of time, which helps track changes in learning performance. The radar graph below shows the distribution of users' answer accuracy under different knowledge topics, which is used to identify strengths and weaknesses.

Figure 4.9 Student's Learning process interface

viii. Recommend Learning Path

Figure 4.11 shows the Recommend Learning Path in learning process page.

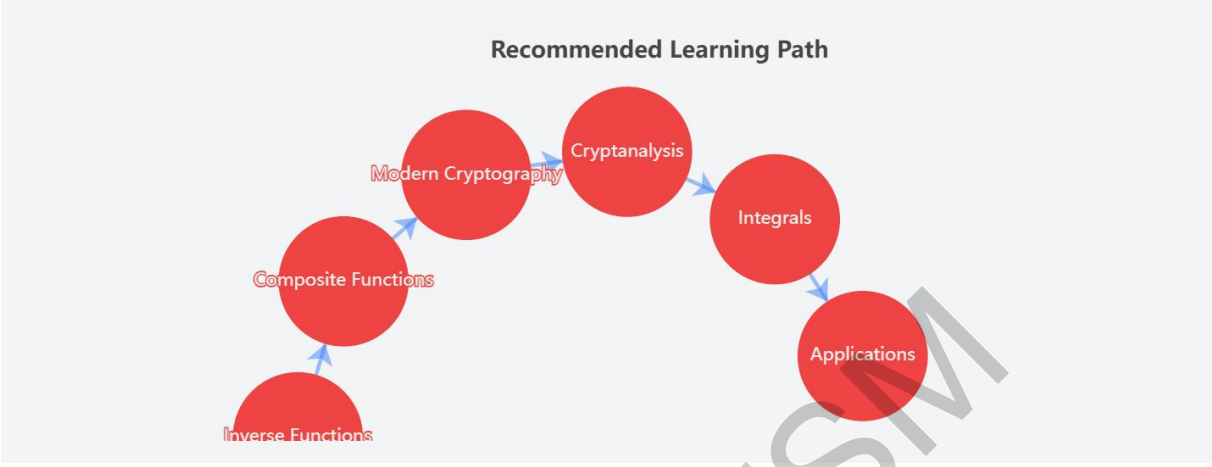

Figure 4.10 Student's Recommend Learning Path

viv. LeaderBoard Interface

The LeaderBoard adopts a left-right double-panel layout: the left side is a collapsible subject-topic navigation tree, and the right side is the ranking list.

Figure 4.11 Student's LeaderBoard Interface

x. Reward Interface

The rewards interface is divided into four functional areas: the Badge Wall displays free achievement badges (colored badges that have been obtained, gray badges that have not been obtained), the Points History displays the details of points obtained, the Task List displays the progress bar and achievable tasks, and the Rewards Store provides paid badge redemption (displays the price and the redemption button). The

two types of badges are distinguished by color and icon, namely free badges and points-purchased badges.

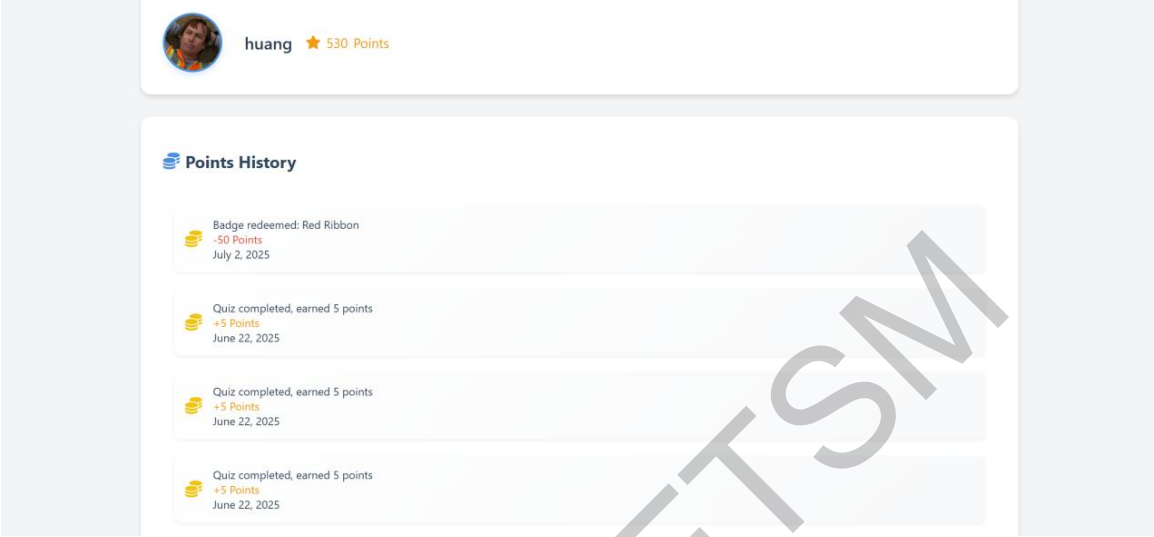

Figure 4.12 Points history in Reward Interface

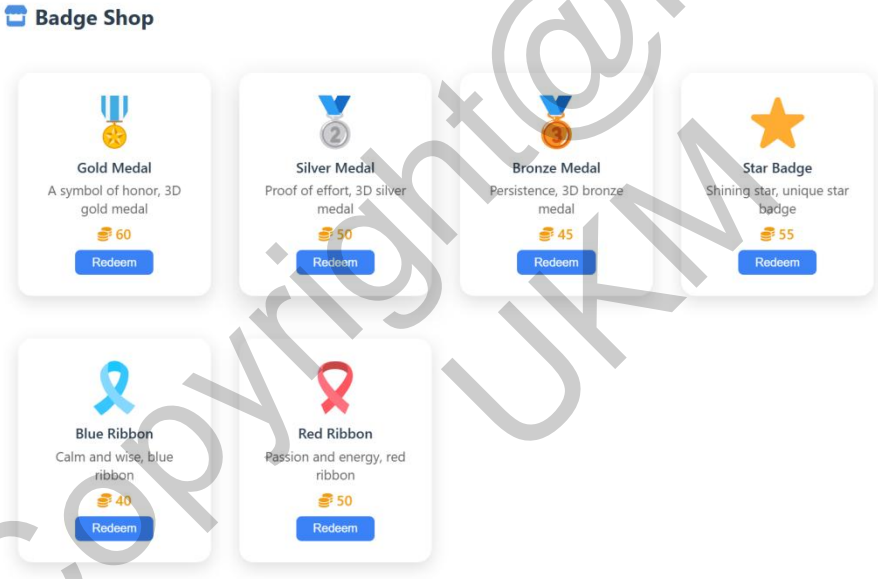

Figure 4.13 Badge Shop in Reward Interface

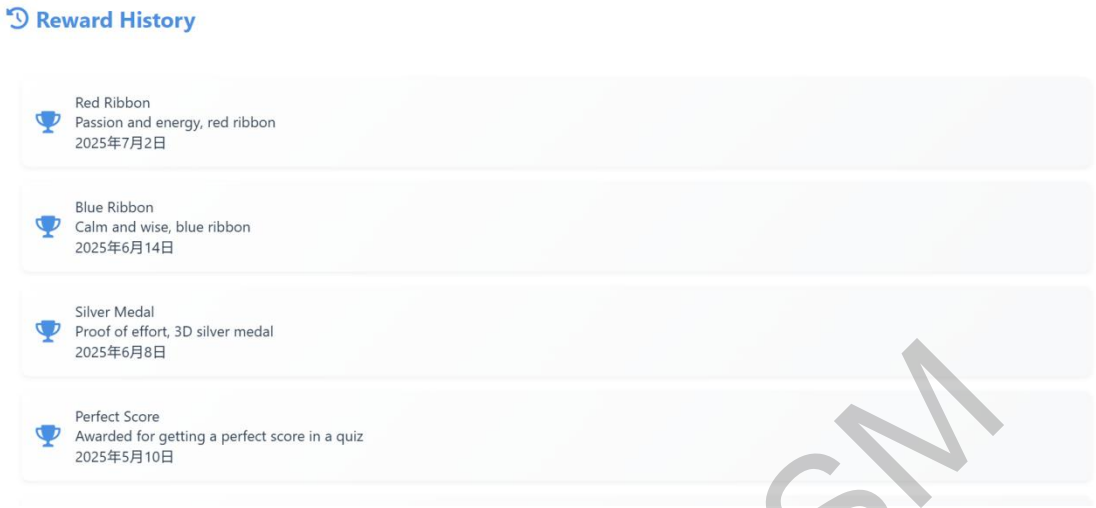

Figure 4.14 Reward history in Reward Interface

xi. Create Quiz Interface

The Quiz creation interface supports multi-choice question editing and provides four colored option input boxes. At the bottom, you can select question type (single choice/multiple choice/short answer), subject, theme, difficulty and other attributes. It has the functions of clearing, saving drafts, and batch publishing, and supports quick switching navigation between questions.

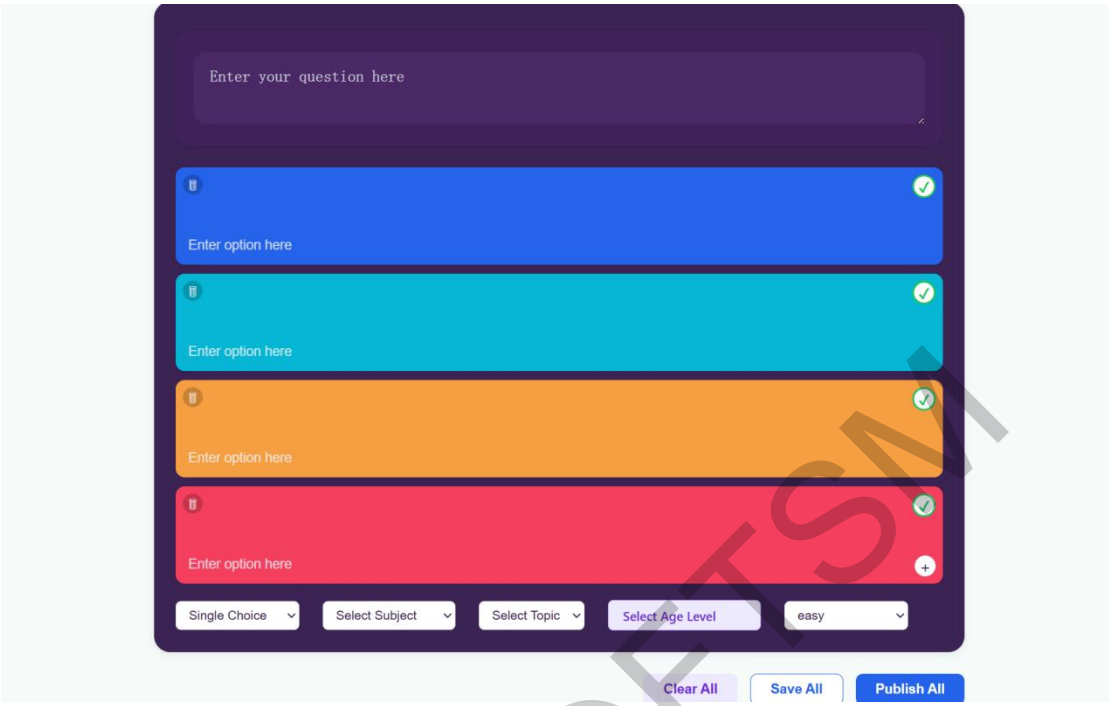

Figure 4.15 Editing interface for multiple-choice or single-choice questions

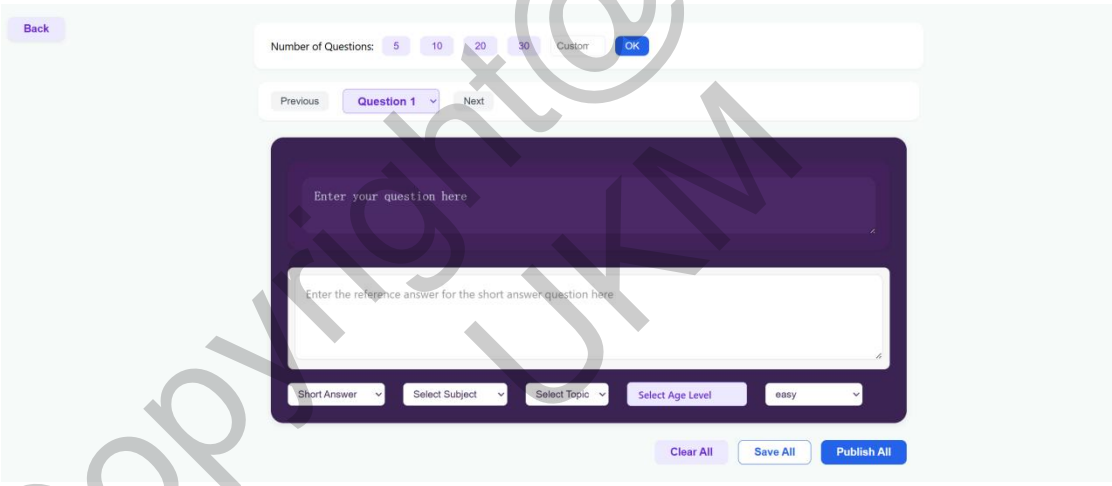

Figure 4.16 Editing interface for short answer questions

xii. Subject Management Interface

This modal box implements the subject management function, including the addition, deletion, modification and query of topics, question screening, learning material upload and other complete teaching content management functions

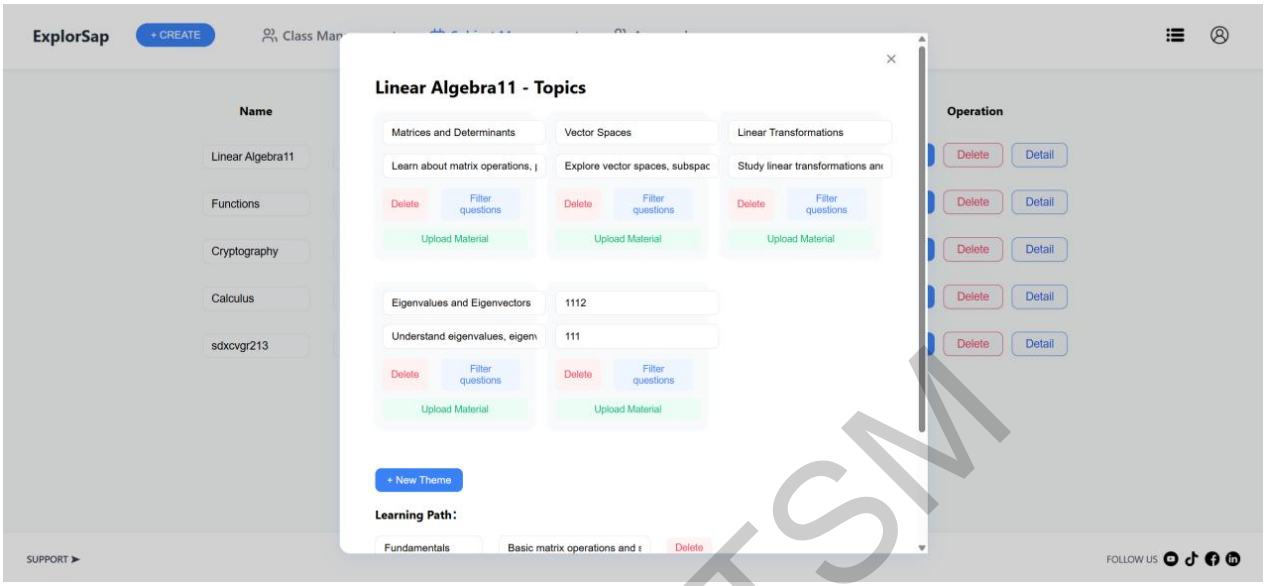

Figure 4.17 Subject Management modal box

xiii. Short Answer Approvals Interface

Figure 4.19 shows the review results (pass/reject) are submitted to the back-end API through asynchronous requests. Upon success, the reviewed items are immediately removed from the interface and the list is refreshed. At the same time, the operation result prompt is displayed, realizing the real-time review and status management of students' short-answer questions by teachers.

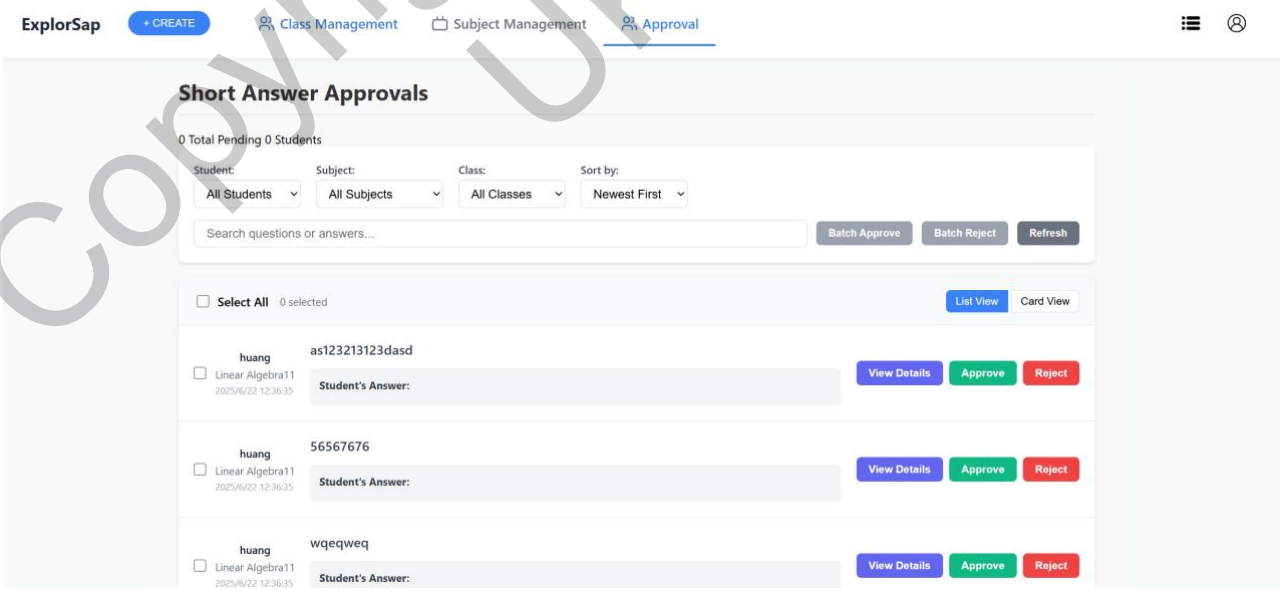

Figure 4.18 Short Answer Approvals Interface

xiv. User Management Interface for Admin

This function implements complete user CRUD operations through dynamic query and paging rendering, including multi-condition screening (user name, role, registration time), batch selection and deletion, password reset, role change and other management functions, supporting real-time search and paging browsing

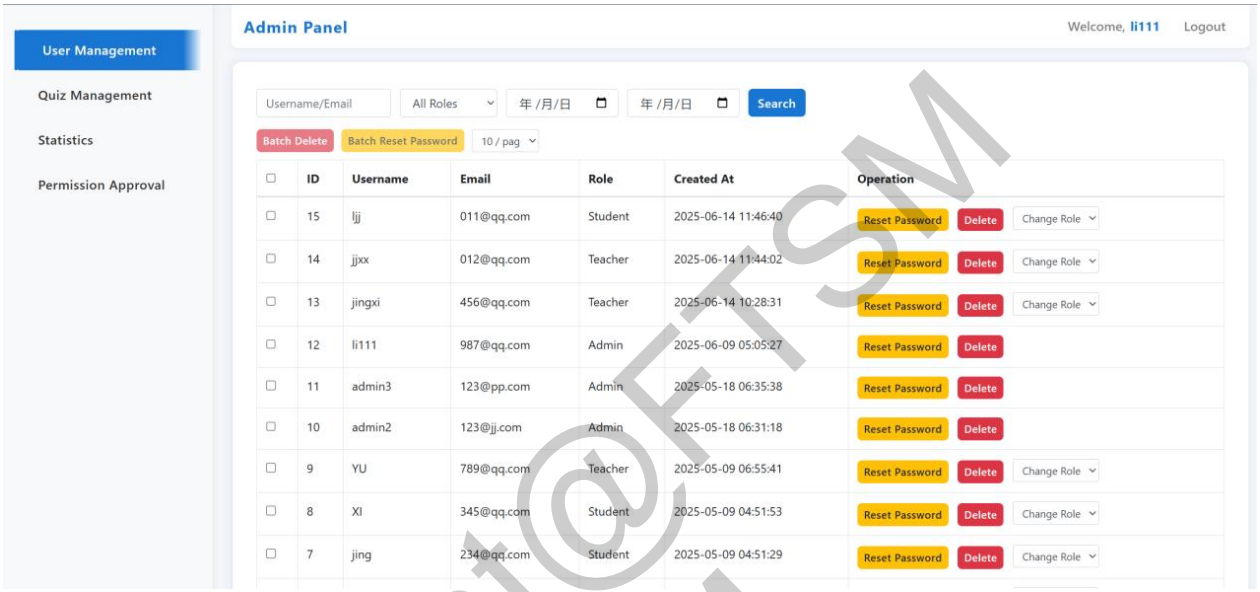

Figure 4.19 User Management Interface

xv. Permission Approval Interface for Admin

Figure 4.21 a shows the the function implements the approval management of teacher permission applications. By obtaining a list of all teacher applications to be reviewed, a table is dynamically rendered to display the applicant's information (user name, email address, application time, status), and a pass/reject button is provided. After clicking, a POST request is sent to the back-end API to update the user status, completing the approval process management of teacher permissions.

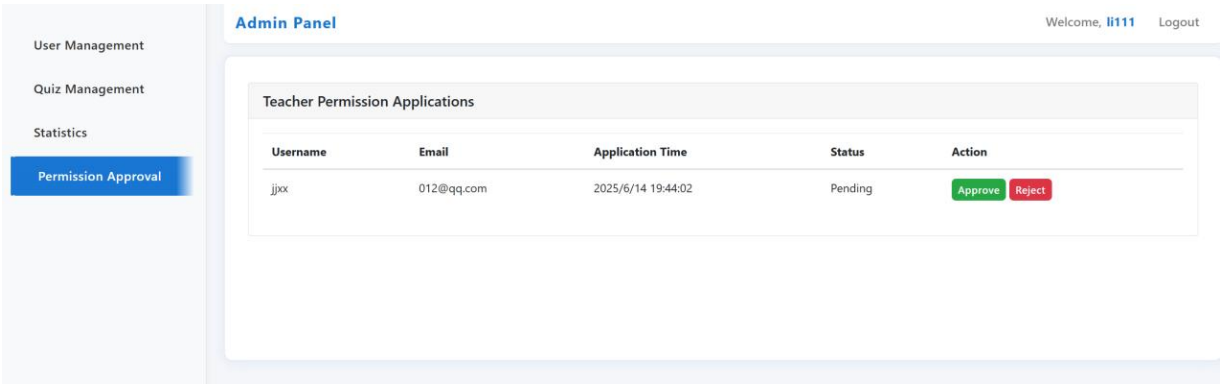

Figure 4.20 Permission Approval Interface

4.3 TESTING IMPLEMENTATION

The testing implementation process was carried out in a systematic manner to ensure that the ExplorSap system meets both functional and non-functional requirements. The testing was divided into multiple stages based on the selected test types, including white box testing, black box (functionality) testing, usability testing, performance testing, and security testing.

4.3.1 Environment Setup

Testing was conducted on the following environment:

a.Front-end: Google Chrome (v123+) using desktop resolution (1920×1080)

b.Back-end: Node.js server running on local machine and Azure cloud instance

c.Database: Azure MySQL

Testing Tools:

a.Postman for API testing (Login, JWT verification, Quiz submission)

b.Browser DevTools for performance timing

c.Console Logging for verifying backend logic

d.Manual User Observation for usability tests

4.3.2 Functional and UAT Testing

Each function module was tested according to its requirement specification and associated test case. The login, quiz participation, leaderboard view, and learning path generation were tested using both valid and invalid data to ensure robustness.

Example:

a.For the Login module, valid and invalid credentials were tested using Postman. JWT token issuance and route protection were verified.

b.For the Quiz module, answer submission was tested by simulating student attempts with different scoring outcomes. System response and accuracy feedback were validated.

c.For the Leaderboard module, user rankings and visibility scope were tested by submitting quiz scores from different accounts.

4.3.3 Usability Testing

The system interface was tested from the perspective of student users. Several students participated in informal sessions where they were asked to complete tasks like:

a.Finding a quiz and submitting answers

b.Viewing their ranking on the leaderboard

c.Navigating to the profile and recommendation sections

d.Feedback was recorded, and UI adjustments (such as button positioning and label clarity) were made accordingly.

4.3.4 Security Testing

Security tests focused on verifying the JWT-based authentication mechanism. Scenarios included:

a.Accessing protected routes without a token

b.Using an expired or modified token

c.Ensuring authMiddleware correctly filters unauthorized requests

d.All security tests resulted in proper HTTP 401 responses, confirming that backend authentication is enforced.

4.3.5 Integration Testing

This integration test involves the collaboration of the following key modules:

a.Content management module - subject and theme creation

b.Question editing module - batch question creation and publishing

c.Gamified learning module - student answering interface and interaction

d.Grading system module - automatic scoring and manual approval

e.Data storage module - multi-table data consistency maintenance

f.Reward system module - points calculation and badge issuance

g.Ranking module - real-time ranking update

4.4 Testing Results

All core modules of the system were tested for functional accuracy using black-box testing. The modules include login, quiz participation, answer validation, leaderboard generation, and personalized learning path recommendation.

Table 4.1 summarize the Functional Testing Results

Test Case ID	Module Tested	Status	Remark
TC_WB_001	Learning Path Recommendation	Pass	-
TC_WB_002	Answer Submission Logic	Pass	-
TC_FN_001	User Login	Pass	-
TC_FN_002	User Verification and Role Management	Pass	-
TC_FN_003	Game Functionality	Pass	-
TC_FN_004	Approval System	Pass	-
TC_FN_005	Reward System	Pass	-
TC_FN_006	Batch Edit Function	Pass	-
TC_FN_007	Subject Management	Pass	-
TC_FN_008	Ranking Function	Pass	-
TC_UAT_001	Leaderboard View	Pass	-
TC_US_001	Sidebar Navigation Usability	Pass	-
TC_SEC_001	JWT Authentication Security	Pass	-

TC_INT_001	publishing-student answering integration test	Pass	-

5.0 CONCLUSION

The ExplorSap education platform uses Node.js and Express.js to build the backend architecture, combined with the Azure MySQL database, to provide a complete education ecosystem for students, teachers and administrators. The system implements core functions such as user authentication and authorization, personalized learning path recommendation, gamified test experience, real-time rankings, reward mechanism, batch question editing, subjective question approval process and multi-dimensional learning analysis.

Through a systematic development process, the ExplorSap platform follows the best practices of software engineering at every stage from demand analysis, system design, implementation development to comprehensive testing. The system architecture adopts a three-tier structure design. The front end uses HTML, CSS and JavaScript to provide a user interface, the back end processes business logic through RESTful API, and the data layer uses a relational database to ensure data consistency and integrity. The entire development process focuses on modular design and code reuse, making the system highly maintainable and scalable.

6.0 REFERENCE

O'Brien, J. A. & Marakas, G. M. 2011. Management Information Systems. New York: McGraw-Hill.

Laudon, K. C. & Laudon, J. P. 2018. Management Information Systems: Managing the Digital Firm. New York: Pearson.

Forster, C., Weiser, C. & Maurer, H. 2015. Emotions and their impact on learning. Learning and Instruction 38: 72-83.

Caponetto, I., Earp, J. & Ott, M. 2014. Gamification in Education: A Systematic

Literature Review. Proceedings of the European Conference on Games-Based Learning: 50-57.

Halverson L.R. 2016. Conceptualising blended learning engagement. Doctoral Dissertation. Brigham Young University, USA.

Halverson L.R. & Graham C.R. 2019. Learner engagement in blended learning environments: A conceptual framework. Online Learning 23(2).

Ma J., Cheng J. & Han X. 2018. Initial development process of a student engagement scale in blended learning environment. Proceedings - 6th International Conference of Educational Innovation through Technology, EITT (2018): 234-237. 10.1109/EITT.2017.63.

Wang A.I. & Lieberoth A. 2016. The effect of narrative elements in digital games on learning. Simulation & Gaming 47(4): 423-444.