

ONLINE EXAMINATION SYSTEM WITH AUTOMATED GRADING

PROJECT REPORT 2025-2026

Submitted By

24507232 YUVARAJ S
24591650 VIJAYLAKSHMI J
22592002 ELAMATHI E
24507217 SANTHI

Under the guidance of

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

LECT/COMPUTER

Diploma in Computer Engineering

of the Directorate of Technical Education, Government of Tamil Nadu



DEPARTMENT OF COMPUTER ENGINEERING

AKT MEMORIAL POLYTECHNIC COLLEGE

KALLAKURICHI-606202.

AKT MEMORIAL POLYTECHNIC COLLEGE

KALLAKURICHI-606202

Department of Computer Engineering

BONAFIDE CERTIFICATE

This is certified that this project work entitled **Online Examination System with Automated Grading** has been submitted **YUVARAJ S, VIJAYLAKSHMI J, ELAMATHI E, SANTHI** in the partial fulfilment of the requirements for the award of Diploma in Computer Engineering during the academic year 2025-2026, who carried out the project work under our supervision.

Project Guide

**Mrs.G.SUGANTHI B.E (CSE),
LECT/COMPUTER**

Head of the Department

**Mrs. C.SRIDEVI B.E.,(CSE)
HOD – COMPUTER & IT**

This is to certify that _____
was examined for the project work viva-voce held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

Acknowledgement

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

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

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

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

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

CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	5
01	INTRODUCTION	6
02	LITERATURE SURVEY	8
03	PROPOSED SYSTEM	11
	3.1 ARCHITECTURE	13
	3.2 MODULES DESCRIPTION	17
04	IMPLEMENTATION DETAILS	22
	4.1 SOFTWARE ENVIRONMENT	25
	4.2 SYSTEM REQUIREMENTS	27
	4.3 SAMPLE CODING	28
	4.4 SCREENSHOT	41
05	CONCLUSION	46
	REFERENCES	48

ABSTRACT

The Online Examination System with Automated Grading using the WordPress WP Survey and Quiz Tool is a modern web-based solution designed to simplify and digitalize the examination process for educational institutions, training centers, and organizations. This system enables administrators and instructors to create, manage, and conduct online examinations efficiently while reducing manual effort and errors associated with traditional evaluation methods.

The platform allows the creation of various types of questions such as multiple-choice, true/false, descriptive, and timed assessments. Students can securely log in, attempt exams within a specified duration, and submit their responses through an intuitive interface. The integration of automated grading ensures that objective-type questions are evaluated instantly, providing immediate feedback and results to students, thereby enhancing transparency and learning outcomes.

Using the WP Survey and Quiz Tool plugin in WordPress, the system benefits from a flexible and user-friendly environment, enabling easy customization, responsive design, and seamless deployment. The system also includes features such as result analytics, performance tracking, leaderboard generation, and report management for administrators.

Security measures like user authentication, time restrictions, randomized questions, and attempt limits ensure the integrity of the examination process. Additionally, the system supports scalability, making it suitable for both small-scale institutions and large online learning platforms.

Overall, this project provides an efficient, cost-effective, and reliable solution for conducting online examinations with automated grading, improving accuracy, saving time, and enhancing the overall assessment experience.

1. INTRODUCTION

The advancement of information and communication technologies has brought a significant transformation in the education sector, shifting traditional teaching and evaluation methods toward digital platforms. One of the most important aspects of education—examinations—has evolved from manual, paper-based systems to automated, online environments. Traditional examination systems often involve challenges such as extensive paperwork, time-consuming evaluation, risk of human errors, difficulty in result processing, and limited scalability. These limitations have created a strong need for a more efficient, accurate, and scalable solution, leading to the development of the Online Examination System with Automated Grading.

This project leverages the capabilities of WordPress, a widely used and flexible content management system, along with the WP Survey and Quiz Tool, to build a dynamic and user-friendly online examination platform. WordPress provides a robust foundation with features like easy content management, plugin integration, responsive design, and security enhancements. The WP Survey and Quiz Tool plugin further extends these capabilities by enabling the creation of customized quizzes, surveys, and assessments without requiring complex coding knowledge.

The system is designed to support multiple user roles such as administrators, instructors, and students. Administrators can manage the entire system, including user accounts, exam settings, and reports. Instructors can create question banks, design exams with different formats (such as multiple-choice questions, true/false, fill-in-the-blanks, and descriptive answers), and define parameters like time limits, attempt restrictions, and grading criteria. Students, on the other hand, can securely log in, attempt exams, and receive instant feedback based on their performance.

A key highlight of this system is its automated grading mechanism. Objective-type questions are evaluated instantly upon submission, eliminating the need for manual correction and significantly reducing the workload on educators. This ensures faster

result processing, improved accuracy, and transparency in evaluation. For subjective questions, the system can either support manual grading by instructors or integrate semi-automated evaluation methods.

In addition to evaluation, the system provides advanced analytical features such as performance tracking, score analysis, and detailed reporting. These insights help educators identify learning gaps, monitor student progress, and improve teaching strategies. Students also benefit from immediate feedback, which enhances their understanding and encourages self-assessment.

Security is a critical aspect of any online examination system. This project incorporates various security measures, including user authentication, encrypted login sessions, randomized question sets, time-bound access, and attempt limitations. These features help maintain the integrity and fairness of examinations while preventing malpractice.

Furthermore, the system is highly scalable and adaptable, making it suitable for a wide range of applications, including schools, colleges, online learning platforms, coaching centers, and corporate training programs. Its web-based nature ensures accessibility from any location, enabling remote examinations and supporting the growing trend of distance education.

In conclusion, the Online Examination System

2. LITERATURE SURVEY

The literature related to online examination systems shows a clear transition from conventional paper-based assessment methods to web-based and automated evaluation platforms. Earlier examination models were largely manual, requiring physical question papers, invigilation, handwritten answer sheets, and time-intensive valuation processes. Researchers and developers have consistently identified major limitations in such systems, including delayed result publication, human calculation errors, high administrative workload, printing cost, and difficulty in conducting exams for a large number of learners. With the growth of internet access and learning management technologies, online examination systems became an important research and implementation area because they offer speed, flexibility, centralized control, and better data management.

Many existing web-based examination systems are designed around core functions such as user authentication, question bank management, timed tests, random question generation, automated scoring, and result reporting. These systems mainly focus on objective assessments, especially multiple-choice, true/false, and fill-in-the-blank questions, because such formats are easier to evaluate instantly through rule-based grading. Studies and practical implementations show that automated grading improves evaluation speed, reduces faculty workload, and increases consistency in score calculation. At the same time, literature also points out that online exams must address reliability issues such as browser compatibility, secure login, network dependency, question leakage, and student malpractice. Because of this, modern systems increasingly include shuffled questions, attempt restrictions, timer controls, and role-based access.

Another important theme in the literature is the use of content management systems and plugin-based development for educational applications. Instead of building an examination platform entirely from scratch, many developers prefer extensible CMS platforms because they reduce development time, simplify administration, and provide reusable modules for authentication, dashboards, and reporting. WordPress is widely

described by its official project as open-source software and a full-featured content management system that can be extended through plugins, making it suitable for rapid web application deployment. This makes WordPress a practical base for academic mini-projects and final-year systems where ease of deployment, maintainability, and cost-effectiveness are important.

Within the WordPress ecosystem, quiz and survey plugins have played a major role in supporting online assessments. The plugin commonly referred to as the WP Survey and Quiz Tool is represented in the WordPress plugin repository by Quiz and Survey Master (QSM). Its official plugin description highlights features such as quiz and survey creation, countdown timers, scheduled availability windows, multi-page quizzes, randomized questions and answers, login requirements, question-bank reuse, reporting, and analysis dashboards. These features align closely with the needs identified in prior online examination literature, particularly the need for controlled assessment environments, automated result generation, and analytic review of learner performance. The presence of reporting tools is especially relevant because recent educational systems are expected not only to conduct tests but also to produce meaningful insights about student performance trends.

Existing literature and practical systems also emphasize the educational value of immediate feedback. Traditional systems often delay feedback until answer scripts are checked and marks are recorded manually. In contrast, automated online examination systems can display scores, correct answers, pass/fail outcomes, and completion statistics immediately after submission. The QSM plugin's official description also notes user dashboard features showing score, time taken, downloadable certificate link, and result page information. This supports a more learner-centered approach in which assessment becomes part of the learning cycle rather than only a grading mechanism. From a pedagogical perspective, such instant response helps students identify weak areas quickly and encourages repeated practice.

Security and fairness remain recurring subjects in the literature survey of online exams. Researchers commonly note that the success of digital assessment depends on user authentication, question randomization, time enforcement, controlled access, and data integrity. Plugin-based systems for WordPress address some of these requirements through login restrictions, scheduled test windows, and randomized questions or answers. While these measures do not completely eliminate malpractice, the literature suggests they significantly improve examination control compared with simple static web forms. For institutional use, additional layers such as HTTPS, role-based permissions, secure hosting, and database backups are also recommended.

The survey of related tools also shows that many platforms exist for forms, surveys, and quizzes, but they differ in purpose and suitability. Some are designed mainly for feedback collection, while others support timed tests, automatic scoring, and exam-style workflows. In comparison, WordPress quiz-oriented tools stand out because they combine ease of content publishing with interactive assessment capabilities. WordPress itself highlights extensibility and plugin-based customization as a core strength. This supports the choice of a WordPress-based examination system for institutions that need a manageable web interface without building an entire platform from the ground up.

From the reviewed literature and available platform features, it is clear that an online examination system built on WordPress with a quiz plugin can satisfy many present-day educational requirements: centralized exam management, automated grading for objective questions, progress tracking, timed assessments, and result analytics. At the same time, the literature indicates a gap between generic survey tools and a more structured examination-oriented implementation tailored for institutional use. Therefore, the proposed project aims to bridge that gap by using WordPress and the WP Survey and Quiz Tool to create a practical, scalable, and academically useful Online Examination System with Automated Grading.

3. PROPOSED SYSTEM

The proposed system is a web-based Online Examination System with Automated Grading developed using WordPress and the WP Survey and Quiz Tool. The system is designed to provide a secure, scalable, and efficient platform for conducting online examinations and evaluating student performance automatically. It aims to eliminate the limitations of traditional examination methods by introducing automation, real-time processing, and centralized management.

The system follows a modular and role-based architecture, where different users such as administrators, instructors, and students interact with the system through dedicated dashboards. The administrator has full control over system configuration, user management, exam scheduling, and report generation. Instructors can create and manage question banks, design exams with multiple question formats (MCQ, true/false, descriptive, etc.), and define parameters such as time limits, marks distribution, and attempt restrictions. Students can log in securely, attend exams, and view their results instantly after submission.

A core component of the proposed system is the automated grading mechanism. Objective-type questions are evaluated instantly by the system using predefined correct answers, which ensures accuracy and reduces manual workload. For subjective questions, the system allows instructor-based evaluation or semi-automated review. This hybrid evaluation approach ensures both efficiency and flexibility in assessment.

The system also incorporates dynamic exam generation, where questions are randomly selected from a large question bank. This reduces the chances of malpractice and ensures fairness among students. Features such as timer-based exams, auto-submit on timeout, question shuffling, and limited attempts further strengthen exam integrity.

To enhance usability and performance monitoring, the system provides real-time analytics and reporting features. These include student performance reports, score analysis, pass/fail statistics, leaderboard generation, and downloadable result

summaries. Such insights help educators make informed decisions and improve teaching strategies, while students can track their progress and identify weak areas.

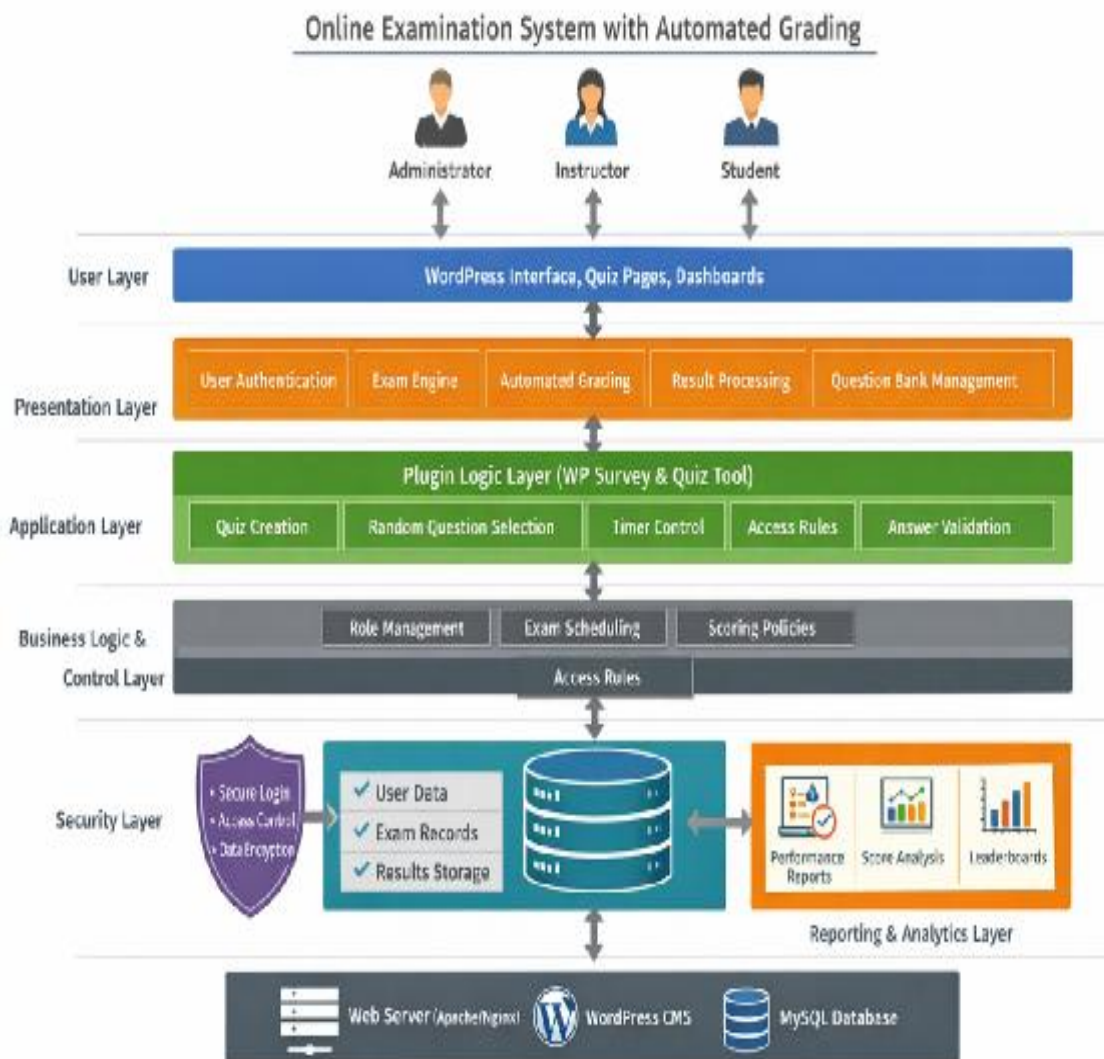
Security is a major focus of the proposed system. It includes user authentication, role-based access control, secure login sessions, and restricted exam access. Exams can be scheduled for specific time windows, and access is limited only to authorized users. Data is stored securely in the backend database, ensuring confidentiality and integrity.

The system is designed with a responsive user interface, making it accessible across desktops, tablets, and mobile devices. Being web-based, it supports remote examinations, allowing users to participate from any location with internet connectivity. This makes the system highly suitable for modern educational environments, including distance learning and online certification programs.

In addition, the proposed system is cost-effective and easy to deploy, as it utilizes the plugin-based architecture of WordPress. This reduces development time and allows easy customization and maintenance. Future enhancements such as AI-based evaluation, proctoring systems, and integration with learning management systems can also be incorporated.

In conclusion, the proposed system offers a comprehensive, automated, and user-friendly solution for conducting online examinations. It improves efficiency, ensures accuracy, enhances security, and provides valuable insights, making it an ideal platform for digital assessment in today's education ecosystem.

3.1 ARCHITECTURE



The architecture of the Online Examination System with Automated Grading is designed as a structured, web-based, multi-layer model that ensures smooth communication between users, application services, plugin logic, and the database. Since the project is developed using WordPress and the WP Survey and Quiz Tool, the architecture combines the flexibility of a content management system with the functional strength of an online examination platform. The complete architecture is organized in such a way that it supports exam creation, student participation, automated grading, result generation, and report analysis within a centralized environment.

At the top level of the architecture is the User Layer, which consists of three main actors: Administrator, Instructor, and Student. Each user accesses the system through a web browser using a secure login page. The administrator manages the complete system, including users, exam schedules, settings, and reports. The instructor is responsible for creating question banks, designing quizzes or examinations, defining grading rules, and reviewing performance. The student interacts with the examination interface, attends the exam, submits answers, and views results. This user layer acts as the entry point to the application and is connected to the presentation layer through role-based authentication.

The next part of the architecture is the Presentation Layer, which includes the front-end pages and dashboards displayed to different users. Since the project is implemented on WordPress, this layer uses WordPress themes, page templates, navigation menus, login forms, quiz pages, result pages, and admin dashboards. The WP Survey and Quiz Tool provides dedicated interfaces for quiz creation, question display, answer submission, timer setup, and result visualization. The presentation layer is responsible for making the system user-friendly, responsive, and accessible across different devices such as desktops, laptops, tablets, and smartphones.

Below the presentation layer is the Application Layer, which handles the core processing of the system. This is the most important layer in the architecture because it controls the actual examination workflow. It includes modules such as user authentication, exam scheduling, question bank management, exam engine, answer validation, automated grading, and result publishing. When a student starts an exam, the application layer fetches the relevant questions, applies rules like timer control and randomization, records the student's responses, and triggers the grading mechanism after submission. For objective-type questions, marks are calculated instantly based on the correct answers stored in the system. For descriptive questions, the application allows manual or semi-automatic evaluation by the instructor. This layer ensures that all exam-related functions are executed accurately and efficiently.

Integrated within the application layer is the Plugin Logic Layer, which is powered by the WP Survey and Quiz Tool. This layer acts as the bridge between WordPress and the examination-specific features. It provides built-in functions for creating quizzes, assigning questions, setting time limits, enabling multiple attempts, shuffling questions, restricting access, and generating score reports. The plugin logic layer simplifies development because it avoids the need to build the entire examination engine from scratch. Instead, it extends WordPress with exam-based capabilities while maintaining compatibility with the WordPress admin panel and user system. This layer is essential because it handles the majority of the online test operations in a structured and reusable way.

The next major part is the Business Logic and Control Layer, which manages system rules and decision-making processes. It ensures that only authenticated users can access their permitted modules, exams are opened only within the scheduled time window, attempt limits are respected, and results are displayed according to administrator settings. It also applies grading policies, pass/fail calculations, score percentages, and result status generation. In this layer, role-based access control is highly important. For example, students can only attend exams and view their own results, instructors can create and manage exam content, and administrators can control the entire system. This layer guarantees secure and proper system behavior.

Beneath this is the Database Layer, which stores all the information required for the examination process. The database contains user records, role details, exam information, question banks, answer keys, student responses, marks, performance history, and generated reports. WordPress primarily uses a MySQL or MariaDB database, and the plugin stores quiz-related data either in WordPress tables or custom plugin tables. Whenever a user logs in, starts an exam, submits answers, or requests a report, the system communicates with the database layer to retrieve or save information. This ensures data persistence, accuracy, and centralized management of exam records.

The architecture also includes a Security Layer, which works across all layers of the system. Security is especially important in an online examination platform because the system deals with sensitive data such as user credentials, exam questions, answers, and results. This layer includes secure login authentication, hashed passwords, session control, access permissions, time-based restrictions, and secure form submission. It may also include features such as CAPTCHA, HTTPS-based communication, and random question delivery to reduce the risk of misuse or cheating. The security layer ensures confidentiality, integrity, and fairness throughout the examination lifecycle.

Another important part of the architecture is the Reporting and Analytics Layer. After exams are completed, the system generates reports such as individual results, pass/fail status, marks obtained, correct and wrong answer counts, overall score percentage, and comparative performance charts. Administrators and instructors can use these reports to monitor student progress and evaluate learning outcomes. Students can also access their personal performance results for self-analysis. This layer improves decision-making and adds academic value to the system beyond simple examination conduction.

Finally, the architecture is supported by a Web Server and Hosting Environment, where WordPress, the plugin, and the database are deployed. This may include Apache or Nginx server, PHP runtime, and MySQL database server. The hosting environment ensures that the application is accessible online and available for remote users. Because the system is browser-based, no special software installation is required on the student side other than an internet browser.

3.2 MODULES DESCRIPTION

1. User Management Module

The User Management Module is responsible for handling all user-related operations within the system. It supports multiple roles such as administrator, instructor, and student, each having specific permissions and access levels. The module enables user registration, secure login, profile management, password reset, and account deactivation. Administrators can create new users manually or allow self-registration depending on system configuration. Role-based access control ensures that users can only access features relevant to their responsibilities.

This module integrates with WordPress authentication mechanisms, ensuring secure session handling and encrypted password storage. It also maintains user activity logs to track login history and system usage. Instructors and students can update their personal details, while administrators can monitor and manage all user accounts centrally. Additionally, the module supports bulk user import, which is useful for institutions managing large numbers of students.

Overall, the User Management Module ensures system security, organized user handling, and efficient access control, forming the foundation for all other modules in the Online Examination System.

2. Admin Dashboard Module

The Admin Dashboard Module acts as the control center of the entire system. It provides administrators with a comprehensive overview of system activities, including the number of users, active exams, completed tests, and performance statistics. The dashboard presents data in a structured and visual format, enabling quick decision-making and monitoring.

Through this module, administrators can configure system settings such as exam rules, grading policies, time limits, and access permissions. It also allows management of plugins, themes, and overall system customization in WordPress. Administrators can track real-time exam activities and identify any irregularities or system issues.

The dashboard includes shortcuts to key functionalities such as user management, question bank, exam creation, and reporting. Notifications and alerts are also displayed to keep administrators informed about important events like upcoming exams or system updates.

In summary, the Admin Dashboard Module simplifies system administration by providing centralized control, real-time insights, and easy navigation across all major functionalities.

3. Question Bank Management Module

The Question Bank Management Module allows instructors to create, organize, and maintain a repository of questions used for examinations. It supports various question types such as multiple-choice, true/false, fill-in-the-blanks, and descriptive questions. Each question can be categorized by subject, topic, difficulty level, and marks allocation.

This module enables instructors to add, edit, delete, and update questions easily. Questions can be stored in a structured format, allowing reuse across multiple exams. Bulk import and export features are also available, making it easier to manage large question sets.

The module supports multimedia-based questions, including images, audio, and videos, enhancing the quality of assessments. Additionally, correct answers and marking schemes are predefined for automated grading.

By maintaining a well-organized question bank, the system ensures efficient exam creation and consistency in assessment. It also supports random question selection, which improves fairness and reduces the chances of cheating.

4. Exam Creation & Management Module

This module is responsible for designing and managing online examinations. Instructors can create exams by selecting questions from the question bank and defining parameters such as exam duration, total marks, number of questions, and passing criteria.

The module allows scheduling of exams with specific start and end times. It also supports randomization of questions and answers to ensure fairness. Instructors can set rules like negative marking, attempt limits, and auto-submit on timeout.

Exams can be previewed before publishing, ensuring accuracy and completeness. Once published, students can access the exam based on their permissions. The module also allows editing or rescheduling exams if needed.

Overall, this module simplifies the process of exam preparation and ensures a structured and controlled assessment environment.

5. Exam Attempt Module (Student Interface)

The Exam Attempt Module provides an interactive interface for students to take exams online. Students can log in securely, view available exams, and start tests within the scheduled time.

The interface displays questions one by one or in a multi-page format, along with a countdown timer. Students can navigate between questions, mark answers, and review their responses before submission. The system automatically saves responses to prevent data loss.

Features such as auto-submit on timeout, warning alerts, and progress indicators enhance the user experience. The module ensures that students can complete exams smoothly without technical difficulties.

This module plays a critical role in delivering a seamless and user-friendly examination experience for students.

6. Automated Grading Module

The Automated Grading Module is one of the most important components of the system. It evaluates objective-type questions instantly based on predefined correct answers stored in the database.

Once a student submits the exam, the system calculates marks automatically and generates results within seconds. This eliminates manual evaluation, reduces errors, and saves time for instructors.

The module supports various grading rules such as positive marking, negative marking, and partial scoring. For subjective questions, it allows manual evaluation by instructors.

Overall, this module ensures accuracy, efficiency, and transparency in the evaluation process, making it a key feature of the system.

7. Result & Report Management Module

This module handles the generation, storage, and display of exam results. After evaluation, results are presented to students in a detailed format, including marks obtained, correct and incorrect answers, and overall performance.

Administrators and instructors can generate reports such as individual performance, class averages, pass/fail statistics, and comparative analysis. Reports can be exported in formats like PDF or Excel for further use.

The module also supports leaderboard generation and ranking systems, motivating students to improve their performance. Historical data is stored for future reference and analysis.

This module enhances transparency and helps both students and educators track academic progress effectively.

8. Security & Authentication Module

The Security Module ensures safe and secure operation of the system. It includes features such as user authentication, password encryption, session management, and role-based access control.

The system restricts unauthorized access and ensures that only eligible users can participate in exams. Features like CAPTCHA, secure login, and HTTPS protocols protect user data.

Exam security is enhanced through random question generation, time restrictions, and attempt limitations. These measures reduce the chances of cheating and maintain fairness.

Overall, this module ensures data confidentiality, integrity, and system reliability.

9. Notification & Communication Module

This module manages communication between the system and users. It sends notifications related to exam schedules, results, login alerts, and system updates. Students receive reminders for upcoming exams, while instructors are notified about submissions and evaluation tasks. Notifications can be delivered via email or dashboard alerts.

This module ensures that users stay informed and engaged with the system, improving overall efficiency and user experience.

10. Analytics & Performance Tracking Module

The Analytics Module provides insights into student performance and system usage. It generates graphical reports showing trends, score distributions, and improvement patterns.

Instructors can analyze individual and group performance to identify strengths and weaknesses. Administrators can monitor system usage and exam success rates. These insights help in improving teaching strategies and decision-making. The module plays a crucial role in enhancing the overall effectiveness of the examination system.

4. IMPLEMENTATION

The implementation of the Online Examination System with Automated Grading is carried out using a web-based development approach, integrating WordPress with the WP Survey and Quiz Tool to create a flexible and scalable examination platform. The system is deployed in a server environment that includes technologies such as PHP for backend processing, MySQL for database management, and Apache or Nginx as the web server. The front-end is developed using HTML, CSS, JavaScript, and responsive WordPress themes to ensure compatibility across devices. Initially, the development environment is configured either locally using tools like XAMPP or directly on a cloud-based VPS server. WordPress is installed and connected to the database, followed by the installation and activation of the WP Survey and Quiz Tool plugin, which provides built-in functionalities for quiz creation, evaluation, and reporting.

The database plays a crucial role in the implementation, as it stores all system data including user details, exam configurations, question banks, responses, and results. WordPress default tables handle user authentication and content management, while the plugin extends the database with additional tables for quiz-related data. Proper database design ensures efficient data storage, retrieval, and integrity. Relationships are established between tables such as users, exams, questions, and results to maintain consistency. Indexing and optimization techniques are applied to improve performance, especially when handling large numbers of users and exam records. Data security is also ensured through secure storage practices and controlled access mechanisms.

User role management is implemented using WordPress's built-in role-based access control system. Three main roles are defined: administrator, instructor, and student. The administrator has complete control over the system, including user management, system settings, plugin configuration, and report generation. Instructors are responsible for creating and managing exams, adding questions, and reviewing performance reports. Students can register, log in, attempt exams, and view their results. Permissions are carefully assigned to ensure that users can only access functionalities relevant to their

roles. This structure enhances system security and prevents unauthorized access or misuse of features.

The question bank and exam creation process form the core functionality of the system. Instructors use the plugin interface to create a repository of questions categorized by subject, topic, and difficulty level. Various question types such as multiple-choice, true/false, fill-in-the-blanks, and descriptive questions are supported. Each question is assigned marks and correct answers, which are used for automated grading. Exams are created by selecting questions from the bank and defining parameters such as duration, total marks, number of attempts, and passing criteria. Additional features such as question randomization, answer shuffling, and time limits are configured to ensure fairness and reduce malpractice. The system also allows previewing exams before publishing, ensuring accuracy and completeness.

During exam execution, students log in to the system and access available exams through their dashboard. Once an exam begins, a timer is activated, and questions are displayed according to the configured format. Students can navigate between questions, select or input answers, and review their responses before submission. The system automatically saves responses periodically to prevent data loss due to technical issues. If the time limit expires, the exam is auto-submitted to ensure fairness. The interface is designed to be user-friendly and responsive, allowing students to take exams on various devices including desktops, tablets, and smartphones.

The automated grading mechanism is implemented to evaluate objective-type questions instantly after submission. The system compares student responses with predefined correct answers stored in the database and calculates marks accordingly. This eliminates manual evaluation, reduces errors, and provides immediate results to students. For subjective questions, instructors can manually review and assign marks through the system interface. The grading module also supports features such as negative marking, partial scoring, and customized grading rules. This hybrid evaluation approach ensures both efficiency and flexibility in assessment.

Once evaluation is completed, the system generates detailed results and reports. Students can view their scores, correct and incorrect answers, and overall performance immediately after submission. Instructors and administrators can access advanced reports including class performance, average scores, pass/fail statistics, and individual progress tracking. These reports can be exported in formats such as PDF or Excel for further analysis. The analytics feature helps educators identify learning gaps, monitor student progress, and improve teaching strategies. Historical data is stored in the database for future reference and comparison.

Security is an essential aspect of the implementation. The system uses secure login authentication, encrypted passwords, and session management to protect user data. Role-based access control ensures that only authorized users can access specific modules. Additional security measures such as HTTPS communication, CAPTCHA verification, and restricted exam access based on time and eligibility are implemented. Randomization of questions and answers further enhances exam integrity by minimizing the chances of cheating. Regular backups and updates are also maintained to ensure system reliability and data protection.

Finally, the system is deployed on a live server, making it accessible to users over the internet. Hosting can be done on a VPS or cloud platform with sufficient resources to handle concurrent users. The deployment process includes configuring the domain, database, and server settings, followed by testing the system for performance, security, and usability. Maintenance activities such as plugin updates, database optimization, and monitoring are carried out regularly to ensure smooth operation.

4.1 SOFTWARE REQUIREMENTS

1. Operating System

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

2. Web Server

- Apache Server (with mod_rewrite enabled)

OR

- Nginx Server

3. Backend Technology

- PHP \geq 8.2
- Laravel Framework (Latest Version)

4. Database

- MySQL Server (5.7 or higher)

OR

- MariaDB

5. Frontend Technology

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

6. Required PHP Extensions

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

- cURL PHP Extension

7. Additional Tools & Software

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

8. Server Configuration

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

9. Browser Compatibility

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Safari

10. Hosting Environment

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

4.2 SYSTEM REQUIREMENTS

1. Hardware Requirements

Development System Requirements

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

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

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

Server Requirements (VPS Hosting)

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

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

4.3 SAMPLE CODING

```
<?php
```

```
global $wpdb;
```

```
if ( !session_id() )
```

```
    session_start();
```

```
define( 'WPSQT_PAGE_MAIN'          , 'wpsqt-menu' );
```

```
define( 'WPSQT_PAGE_QUIZ'          , 'wpsqt-menu-quiz' );
```

```
define( 'WPSQT_PAGE_QUESTIONS'    , 'wpsqt-menu-question' );
```

```
define( 'WPSQT_PAGE_QUIZ_RESULTS'  , 'wpsqt-menu-quiz-results' );
```

```
define( 'WPSQT_PAGE_OPTIONS'       , 'wpsqt-menu-options' );
```

```
define( 'WPSQT_PAGE_HELP'          , 'wpsqt-menu-help' );
```

```
define( 'WPSQT_PAGE_SURVEY'        , 'wpsqt-menu-survey' );
```

```
define( 'WPSQT_PAGE_CATN'          , 'wpsqt-menu-catn' );
```

```
define( 'WPSQT_PAGE_MAINTENANCE'   , 'wpsqt-menu-maintenance' );
```

```
define( 'WPSQT_TABLE_QUIZ_SURVEYS' , $wpdb-
```

```
>get_blog_prefix(). 'wpsqt_quiz_surveys' );
```

```
define( 'WPSQT_TABLE_SECTIONS'     , $wpdb-
```

```
>get_blog_prefix(). 'wpsqt_sections' );
```

```
define( 'WPSQT_TABLE_QUESTIONS'   , $wpdb-
```

```
>get_blog_prefix(). 'wpsqt_all_questions' );
```

```
define( 'WPSQT_TABLE_FORMS'        , $wpdb-
```

```
>get_blog_prefix(). 'wpsqt_custom_forms' );
```

```
define( 'WPSQT_TABLE_RESULTS'      , $wpdb-
```

```
>get_blog_prefix(). 'wpsqt_all_results' );
```

```

define(      'WPSQT_TABLE_SURVEY_CACHE'          ,      $wpdb-
>get_blog_prefix().'wpsqt_survey_cache_results' );
define(      'WPSQT_TABLE_QUIZ_STATE'            ,      $wpdb-
>get_blog_prefix().'wpsqt_quiz_state' );

define(      'WPSQT_URL_MAIN'                    ,
admin_url('admin.php?page='.WPSQT_PAGE_MAIN) );
define(      'WPSQT_URL_MAINTENANCE'            ,
admin_url('admin.php?page='.WPSQT_PAGE_MAINTENANCE) );
define( 'WPSQT_CONTACT_EMAIL'      , 'support@catn.com' );
define( 'WPSQT_VERSION'            , '2.14.1' );
define( 'WPSQT_DIR'                , realpath(dirname(__FILE__)).'/' );
define( 'WPSQT_FILE'               , __FILE__ );

require_once WPSQT_DIR.'lib/Wpsqt/Core.php';
require_once WPSQT_DIR.'lib/Wpsqt/System.php';

// Call Wpsqt_Installer Class to write in WPSQT tables on activation
register_activation_hook ( __FILE__ , 'wpsqt_main_install' );

SoldVersion = get_option('wpsqt_version');
update_option('wpsqt_version',WPSQT_VERSION);
if ( !get_option('wpsqt_number_of_items') ){
    update_option('wpsqt_number_of_items',25);
}
// Simple way of checking if an it's an update or not.
if ( !empty($SoldVersion) && (version_compare($SoldVersion, WPSQT_VERSION) <
0) ){
    require_once WPSQT_DIR.'lib/Wpsqt/Page/Maintenance/upgradeScript.php';
}

```

```

// Make sure admin has the capability
$role = get_role('administrator');
$role->add_cap('wpsqt-manage');

/**
 * Class for Installing plugin on activation.
 *
 * @since 2
 */
function wpsqt_main_install(){

    global $wpdb;

    $wpdb->query("CREATE TABLE IF NOT EXISTS
''.WPSQT_TABLE_QUESTIONS." (
        `id` int(11) NOT NULL AUTO_INCREMENT,
        `name` varchar(512) NOT NULL,
        `type` varchar(255) NOT NULL,
        `item_id` int(11) NOT NULL,
        `section_id` int(11) NOT NULL,
        `difficulty` varchar(255) NOT NULL,
        `order` int(11) DEFAULT NULL,
        `meta` longtext NOT NULL,
        `timestamp` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
        PRIMARY KEY (`id`)
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8
    AUTO_INCREMENT=1 ;");

    $wpdb->query("CREATE TABLE IF NOT EXISTS
''.WPSQT_TABLE_RESULTS."(

```

```

`id` int(11) NOT NULL AUTO_INCREMENT,
`item_id` int(11) NOT NULL,
`datetaken` varchar(255) NOT NULL,
`timetaken` int(11) NOT NULL,
`person` longtext NOT NULL,
`sections` longtext NOT NULL,
`person_name` varchar(255) NOT NULL,
`ipaddress` varchar(255) NOT NULL,
`timestamp` timestamp NOT NULL DEFAULT '0000-00-
00 00:00:00' ON UPDATE CURRENT_TIMESTAMP,
`status` varchar(255) NOT NULL DEFAULT 'unviewed',
`score` INT NULL ,
`total` INT NULL ,
`percentage` INT NULL,
`pass` BOOLEAN NOT NULL,
`cached` TINYINT(1) DEFAULT '0',
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
AUTO_INCREMENT=1 ;");

```

```

$wpdb->query("CREATE TABLE IF NOT EXISTS
`.WPSQT_TABLE_FORMS.` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`item_id` int(11) NOT NULL,
`name` varchar(255) NOT NULL,
`type` varchar(255) NOT NULL,
`required` varchar(255) NOT NULL,
`validation` varchar(355) NOT NULL,
`timestamp` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
PRIMARY KEY (`id`)

```

```
        ) ENGINE=InnoDB DEFAULT CHARSET=utf8
AUTO_INCREMENT=1 ;");
```

```
$wpdb->query("CREATE TABLE IF NOT EXISTS
`.WPSQT_TABLE_QUIZ_SURVEYS.` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `name` varchar(512) NOT NULL,
    `settings` longtext NOT NULL,
    `type` varchar(266) NOT NULL,
    `timestamp` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
AUTO_INCREMENT=1 ;");
```

```
$wpdb->query("CREATE TABLE IF NOT EXISTS
`.WPSQT_TABLE_SECTIONS.` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `item_id` int(11) NOT NULL,
    `name` varchar(255) NOT NULL,
    `limit` varchar(255) NOT NULL,
    `order` varchar(11) NOT NULL,
    `difficulty` varchar(255) NOT NULL,
    `timestamp` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    UNIQUE KEY `id` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
AUTO_INCREMENT=1 ;");
```

```
$wpdb->query("CREATE TABLE IF NOT EXISTS
`.WPSQT_TABLE_SURVEY_CACHE.` (
```

```

        `id` int(11) NOT NULL AUTO_INCREMENT,
        `sections` longtext NOT NULL,
        `total` int(11) NOT NULL,
        `item_id` int(11) NOT NULL,
        `timestamp` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
        PRIMARY KEY (`id`)
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8
AUTO_INCREMENT=1 ;");

```

```

    $wpdb->query("CREATE TABLE IF NOT EXISTS
'".WPSQT_TABLE_QUIZ_STATE.'" (
        `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
        `uid` mediumtext,
        `answers` text,
        `post` text,
        `quiz_id` int(11) DEFAULT NULL,
        `current_section` int(11) DEFAULT NULL,
        PRIMARY KEY (`id`)
    ) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT
CHARSET=utf8;");
}
if (is_admin()){
    if (is_multisite() && get_option('wpsqt_manual') != 1) {
        echo '<div class="error">WPSQT is not fully compatible with multisite
installations. You will need to create the database tables <a
href="'.WPSQT_URL_MAINTENANCE.'&section=debug">manually</a>.</div>';
    }
    require_once WPSQT_DIR.'lib/Wpsqt/Admin.php';
    $objWpsqtPlugin = new Wpsqt_Admin();
} else {

```

```
$objWpsqtPlugin = new Wpsqt_Core();  
}
```

PDF Conversion

```
<?php
```

```
    /**  
     * Handles the fetching and downloading of PDFs  
     * from DocRaptor.  
     *  
     * @author Iain Cambridge  
     * @copyright Fubra Limited 2010-2011 (c), all rights reserved.  
     * @license GPL v2  
     * @package WPSQT  
     */  
  
require_once dirname(dirname(dirname(dirname(__FILE__)))) . '/wp-load.php';  
require_once WPSQT_DIR . 'lib/Wpsqt/Tokens.php';  
require_once WPSQT_DIR . 'lib/docraptor.php';  
  
$resultId = filter_input(INPUT_GET, 'id');  
$quizId = filter_input(INPUT_GET, 'quizid');  
if ( (!$_GET['quizid'] || !$_GET['id'] ) ) {  
    wp_die('No result id given');  
}  
  
global $wpdb;  
  
if ( filter_input(INPUT_GET, 'html') ) {
```

```

        echo $pdfTemplate;
    exit;

} else {

    $quizDetails = $wpdb->get_row(
        $wpdb->prepare("SELECT * FROM
        '".WPSQT_TABLE_QUIZ_SURVEYS.'" WHERE id = %d", array($_GET['quizid'])),
        ARRAY_A);

    $quizDetails['settings'] = unserialize($quizDetails['settings']);

    $resultDetails = $wpdb->get_row(
        $wpdb->prepare("SELECT * FROM
        '".WPSQT_TABLE_RESULTS.'" WHERE id = %d", array($_GET['id'])),
        ARRAY_A );

    $resultDetails['person'] = unserialize($resultDetails['person']);
    $resultDetails['sections'] = unserialize($resultDetails['sections']);

    $personName = ( isset($resultDetails['person']['name']) &&
    !empty($resultDetails['person']['name']) ) ? $resultDetails['person']['name'] :
    'Anonymous';

    $timestamp = strtotime($resultDetails['timestamp']);

    $pdfTemplate = (empty($quizDetails['settings']['pdf_template']) ?
    get_option('wpsqt_pdf_template'):$quizDetails['settings']['pdf_template'];

    if ( empty($pdfTemplate) ){
        // default pdf template here.
        $pdfTemplate = "<html>";
    }
}

```

```

        $pdfTemplate .= "<body>";
        $pdfTemplate .= " <center>You %USER_NAME% passed the
%QUIZ_NAME% quiz!</center>";
        $pdfTemplate .= "</body>";
        $pdfTemplate .= "</html>";
    }

    $objTokens = Wpsqt_Tokens::getTokenObject();
    $objTokens->setDefaultValues();
    $pdfTemplate = $objTokens->doReplacement($pdfTemplate);

    $resultUrl = esc_html(get_bloginfo('url').'/wp-admin/admin.php?page=wpsqt-
menu&type=quiz&action=results&id='.$quizId
        . '&subaction=mark&subid='.$resultDetails['id']);
    $pdfTemplate = str_ireplace('%RESULT_URL%', $resultUrl, $pdfTemplate);
    $url =
plugins_url('pdf.php?html=true&id='.$_GET['id'].'&quizid='.$_GET['quizid'], __FILE
__);
    $sapiKey = get_option('wpsqt_docraptor_api');

    if ( !$sapiKey ){
        print "No DocRaptor API key! Please alert the site owner to fix this!";
        exit;
    }

    $objDocraptor = new DocRaptor($sapiKey);
    $objDocraptor->setDocumentType('pdf')
        ->setName('PDF')
        ->setDocumentContent($pdfTemplate);
    header('Content-disposition: attachment; filename=document.pdf');
    header('Content-type: application/pdf');

```

```

        print $objDocraptor->fetchDocument();

    }

<?php

/**
 * DocRaptor
 *
 * @author Warren Krewenki
 */
class DocRaptor {

    protected $api_key;
    protected $document_content;
    protected $document_url;
    protected $document_type;
    protected $name;
    protected $test;

    public function __construct($api_key=null){
        if(!is_null($api_key)){
            $this->api_key = $api_key;
        }
        $this->test = false;
        $this->setDocumentType('pdf');
        return true;
    }
}

```

```
public function setAPIKey($api_key=null){
    if(!is_null($api_key)){
        $this->api_key = $api_key;
    }
    return $this;
}
```

```
public function setDocumentContent($document_content=null){
    $this->document_content = $document_content;
    return $this;
}
```

```
public function setDocumentUrl($document_url){
    $this->document_url = $document_url;
    return $this;
}
```

```
public function setDocumentType($document_type){
    $document_type = strtolower($document_type);
    $this->type = ($document_type == 'pdf' || $document_type == 'xls') ?
$document_type : 'pdf';
    return $this;
}
```

```
public function setName($name){
    $this->name = $name;
    return $this;
}
```

```
public function setTest($test=false){
    $this->test = (bool)$test;
}
```

```

        return $this;
    }

    public function fetchDocument($filename = false){
        if($this->api_key != ""){
            $url    =    "https://docraptor.com/docs?user_credentials=".$this-
>api_key;

            $fields = array(
                'doc[document_type]'=>$this->type,
                'doc[name]'=>$this->name,
                'doc[test]'=>$this->test
            );
            if ( !empty($this->document_content) ){
                $fields['doc[document_content]']    =    urlencode($this-
>document_content);
            } else {
                $fields['doc[document_url]']        =    urlencode($this-
>document_url);
            }
            $fields_string = "";
            foreach($fields as $key=>$value) { $fields_string .=
$key.'='.$value.'&; }
            $fields_string = rtrim($fields_string, '&');
            $ch = curl_init();
            curl_setopt($ch,CURLOPT_URL,$url);
            curl_setopt($ch,CURLOPT_POST,count($fields));
            curl_setopt($ch,CURLOPT_POSTFIELDS,$fields_string);
            curl_setopt($ch,CURLOPT_RETURNTRANSFER,true);
            if($result = curl_exec($ch)) {
                if($filename){
                    file_put_contents($filename,$result);
                }
            }
        }
    }
}

```

```
        }
    } else {
        echo 'error';
    }

    //close connection
    curl_close($ch);
    return $filename ? true : $result;
}

}

?>
```

4.4. SCREEN SHOT

The screenshot shows the homepage of the A.K.T Memorial Polytechnic College online examination system. The browser address bar shows 'aktprojects.one'. The page features a red header with the text 'WELCOME TO AKT MCQ ONLINE EXAMINATION'. Below the header is a navigation menu with links for HOME, MCQ, REGISTRATION, LOGIN, and LOGOUT. The main content area is titled 'HOME' and contains 'Instructions to Candidates' on the left and a registration form on the right. The instructions include: 'Remember your username and password; if forgotten, you will not be eligible to proceed.', 'The test should be 60 minutes long with 30 questions related to job titles.', 'Each question is worth 1 marks, and the minimum passing score is 60% to advance to the next round.', 'Once you complete the test, click the 'submit' button and await the results.', and 'If the 60 minutes elapse without submission'. The registration form asks for 'Username *' (with 'adminakt' entered), 'Full Name *', and 'Degree' (with 'B DIT' selected from a dropdown). A note says 'Please fill in the form below to register.'

This screenshot is similar to the one above but shows a dropdown menu open over the 'Username *' field. The dropdown menu lists several subjects: ARTIFICIAL INTELLIGENCE, CLOUD COMPUTING, SYSTEM ADMINISTRATION, INFORMATION TECHNOLOGY ESSENTIALS, RDBMS, and DLD. The 'Username *' field contains 'adminakt', 'Full Name *' is empty, and 'Degree' is 'B DIT'. The navigation menu and header are the same as in the previous screenshot.

The screenshot shows a web browser window with the URL `aktprojects.com/artificial-intelligence/`. The page header includes the logo for A.K.T MEMORIAL Polytechnic College and a navigation menu with links for HOME, MCQ, REGISTRATION, LOGIN, and LOGOUT. A red banner at the top right says "WELCOME TO AKT MCQ ONLINE EXAMINATION". Below this, a black bar displays "ARTIFICIAL INTELLIGENCE UI". The main content area features the title "Artificial intelligence Unit 1" and a timer indicating "Time Left: 59 minutes and 59 seconds". Two multiple-choice questions are visible:

1. Why do problem-solving agents use heuristics?
 - To increase complexity
 - To reduce accuracy
 - To simplify problem-solving
 - To avoid solutions
2. What is rationality in AI agents?
 - Acting to maximize goal achievement
 - Reducing system efficiency
 - Making random decisions

The Windows taskbar at the bottom shows the current time as 07:58 PM on 21-03-2024.

This screenshot shows the same website after the exam is completed. The main heading is "Exam Finished" followed by "Artificial intelligence Unit 1". The first question, "Why do problem-solving agents use heuristics?", is displayed in red text with a red 'X' next to it, indicating an incorrect answer. Below it, the correct options are listed:

1. To increase complexity - Given
2. To reduce accuracy
3. To simplify problem-solving
4. To avoid solutions

The second question, "What is rationality in AI agents?", is also shown in red text with a red 'X' next to it, indicating it was not answered or was incorrect.

aktprojects.one/artificial-intelligence/

WELCOME TO AKT MCQ ONLINE EXAMINATION

A.K.T MEMORIAL
Polytechnic College

HOME MCQ REGISTRATION LOGIN LOG-OUT

4. To reduce accuracy

Which type of AI is most prevalent today? X

1. Super AI - Given
2. **Narrow AI**
3. General AI
4. Emotional AI

Total Points: 8 correct out of 30

© COPYRIGHT DESIGNED BY DEPARTMENT OF INFORMATION TECHNOLOGY

WP Survey And Quiz Tool - Survey/Quizzes

All (20) | Quizzes (20) | Surveys (0) | Polls (0)

ID	Title	Status	Results
7	II_CSE_IT_AI_U1	Enabled	121
8	II_DCOMP_CC_UT	Enabled	56
10	II_DCOMP_CC_U2	Enabled	0
11	II_CSE_IT_AI_U2	Enabled	82
13	II_IT_SA_UT	Enabled	62
14	II_CSE_IT_RDBMS_U1	Enabled	94
15	II_IT_JTE_U1	Enabled	15

aktprojects.one/wp-admin/admin.php?page=wpsqt-menu§ion=edit&subsection=quiz&id=7

AKT Online Examination Protection 5 Now Register WPSQT AccelerateWP Howdy, darshan

Create Online Examination

IL_CSE_IT_AI_UI : edit | Sections | Questions | Form | Results Shortcode: [wpsqt name="IL_CSE_IT_AI_UI" type="quiz"] [Insert into New Page](#)

Name What you would like the quiz to be called.

Limit to one submission per IP No Yes Limit the quiz to one submission per IP.

Limit to one submission per WP user No Yes Limit the quiz to one submission per WP user. You must have the Use WP Details option below set to yes.

Limit to one submission per computer (using cookies) No Yes Limit the quiz to one submission per computer/browser.

Allow save/resume No Yes Allow save and resume for this quiz?

Display question numbers? No Yes Select whether you want the numerical question numbers to be displayed next to the text of the question.

Timer value for the quiz Enter the countdown timer value in minutes for the quiz. Enter 0 for no timer.

IL_CSE_IT_AI_UI

Field Display What should be displayed on the finishing of the quiz.

Status What is the quiz either enabled where users can take it or disabled where users can't.

Send notification email to user as well No Yes Should we send a notification email to the user who took the quiz. You must enable the use wordpress details option below and the user must be logged in for this to work. This is due to a bug in the take contact details option.

Take contact details No Yes This will show a form for users to enter their contact details before proceeding.

Use Wordpress user details No Yes This will allow you to have the quiz to use the details of the user if they are signed in. If enabled the contact form will not be shown if enabled.

Complete Notification Send a notification email on completion the quiz by a user.

Custom Email Template The template of the email sent on notification, if empty the default one will be used. [Click here](#) to see the tables that can be used.

PDF Template The template for the PDF, if you're not using pdf certificates then leave blank. [Click here](#) to see the tables that can be used.

PDF Certificates No Yes Allow the user to download a PDF certificate.

5. CONCLUSION

The Online Examination System with Automated Grading developed using WordPress and the WP Survey and Quiz Tool successfully demonstrates an efficient, scalable, and user-friendly solution for modern digital assessments. The system effectively replaces traditional paper-based examinations with a fully automated, web-based platform that simplifies exam management, reduces manual workload, and enhances accuracy in evaluation.

Through its modular design, the system supports multiple user roles including administrators, instructors, and students, ensuring organized and secure access to functionalities. The automated grading feature significantly improves the speed and reliability of result processing by instantly evaluating objective-type questions. This not only saves time for educators but also provides immediate feedback to students, enhancing their learning experience and engagement.

The integration of features such as question bank management, exam scheduling, timer-based assessments, randomized questions, and detailed reporting ensures a comprehensive examination environment. Additionally, the system's security mechanisms—including authentication, role-based access control, and exam restrictions—help maintain fairness and integrity throughout the examination process.

Another key advantage of the system is its flexibility and ease of deployment. By leveraging WordPress and plugin-based development, the project minimizes development complexity while allowing easy customization and scalability. It can be effectively used in schools, colleges, coaching centers, and corporate training programs, supporting both small-scale and large-scale examinations.

In conclusion, this project provides a reliable and cost-effective solution for conducting online examinations with automated grading. It enhances efficiency, ensures transparency, improves data management, and aligns with the growing demand for

digital education systems. Future enhancements such as AI-based evaluation, online proctoring, and integration with learning management systems can further improve its capabilities and make it more advanced and intelligent.

REFERENCES

1. WordPress Official Website.
Available at: <https://wordpress.org>
2. WP Survey and Quiz Tool (Quiz and Survey Master Plugin).
Available at: <https://wordpress.org/plugins/quiz-master-next/>
3. Pressman, R. S. Software Engineering: A Practitioner's Approach. McGraw-Hill Education, 7th Edition.
4. Sommerville, I. Software Engineering. Pearson Education, 10th Edition.
5. Kothari, C. R. Research Methodology: Methods and Techniques. New Age International Publishers.
6. Tanenbaum, A. S., & Van Steen, M. Distributed Systems: Principles and Paradigms. Prentice Hall.
7. Elmasri, R., & Navathe, S. B. Fundamentals of Database Systems. Pearson Education.
8. Welling, L., & Thomson, L. PHP and MySQL Web Development. Addison-Wesley.
9. Mozilla Developer Network (MDN). Web Development Documentation.
Available at: <https://developer.mozilla.org>
10. MySQL Documentation. Oracle Corporation.
Available at: <https://dev.mysql.com/doc/>
11. Apache HTTP Server Documentation. Apache Software Foundation.
12. Available at: <https://httpd.apache.org/docs/>
Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures (REST Dissertation).
13. ISO/IEC 9126. Software Engineering — Product Quality Standard.
14. IEEE. IEEE Standard for Software Requirements Specifications (IEEE 830).
15. Google Developers. Web Security and Best Practices.
Available at: <https://developers.google.com>