iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.59818

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

TESTIFY- Development of a Test Generator Website Using PHP and SQL Database

Maneet Singh[1], Nitesh Sahu[2], Shriyansh Tiwari[3], Mitali Gupta[4], Samiksha Shukla[5]
[1, 2, 3, 4]*B.Tech student,* [5]*Assistant Professor, Department of Information Technology, Government Engineering College Bilaspur*

Abstract: ***This study describes the creation of a test generating website with SQL database management and PHP server-side scripting. The website offers a quick and effective way for educators to produce and deliver exams by giving them a platform to create, organize, and administer tests online. The website's architecture is covered in the paper, along with how PHP is used for server-side functionality, My SQL is used for the database management system, and HTML, CSS, and JavaScript are used for the front-end user interface.***
Additionally, it examines the website's features, including question bank administration, test result generating, test creation, and user authentication. The paper's conclusion includes a discussion of the website's possible effects on education and recommendations for improvements in the road.
Keywords: Test generator, PHP, SQL database, web development, education, online testing

I. INTRODUCTION

A. Overview of online testing

The process of giving exams or evaluations online is referred to as "online testing." It enables people to take exams without physically being in a classroom, usually from the comfort of their own laptops or other devices. There are several uses for online testing, including surveys, employment assessments, certification exams, and educational exams.
Online testing has several benefits, two of which are its accessibility and convenience. Examiners are not required to visit to a particular area because they can do exams from any location with an internet connection. Online testing platforms can also provide features that improve the efficiency and fairness of the testing process, like automatic scoring, instant feedback, and the capacity to randomly generate questions.

B. Purpose of the test generator website

A test generator website is made to make exams, quizzes, and assessments quickly and simply. Educators, trainers, and organizations utilize these tools to create personalized assessments for a range of uses, including staff training, certification exams, and student assessments.
Simplifying the test creation process is the primary goal of a test generator website. Question banks, which let users save and arrange questions for later use, and the option to randomly generate tests with different question and answer options for every test taker are typical features found on these websites.
Furthermore, formatting and customization options for tests, like time restrictions, scoring criteria, and instructions, are frequently included on test generator websites.

C. Significance of using PHP and SQL for web development

In web development, PHP (Hypertext Preprocessor) and SQL (Structured Query Language) are frequently combined for a number of reasons:
Dynamic Content: By directly integrating PHP code into HTML, developers can use PHP to create dynamic, interactive webpages.
Database Integration: SQL is utilized for database management and querying. Developers can store and retrieve data from a database, including user and product details, by utilizing PHP and SQL queries to connect to a database.
Scalability: Because PHP and SQL are both very scalable technologies, they may be used to create websites and online apps that must manage massive volumes of data and traffic.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

II. SYSTEM ARCHITECTURE

A. Front-end technologies (HTML, CSS, JavaScript)

The content of web pages is organized using HTML (Hypertext Markup Language). HTML is used to design the test page layout on an online testing website. This includes where questions and answers are placed as well as any other components like progress indicators or timers.

Cascade Style Sheets, or CSS, are used to style HTML elements to create a visually appealing and user-friendly test-taking environment. The test pages' layout, color scheme, font, and spacing are all defined by CSS, guaranteeing a unified appearance and feel throughout the website.

The test-taking website has dynamic behavior and interactivity thanks to JavaScript. JavaScript can be used in an online test to manage timers or other time-related features, compute scores, validate answers, and give users feedback. Additionally, the website can be made responsive with JavaScript, so that it functions properly on a variety of screens and devices. Together, these front-end technologies produce an easy-to-use and captivating online testing experience for users.

B. Back-end technology (PHP)

1) User authentication: To guarantee that only authorized users can access the test-taking capability, users can be authenticated using PHP. Verifying user credentials against a database of registered users may be one way to accomplish this. Test Management and Generation: PHP can be used to create tests on the fly using pre-set parameters or templates. Test questions, answers, and other relevant data can be stored and retrieved from a database with its help.

2) Calculating Scores and Results: PHP has the ability to compute scores using both user input and pre-established answer keys. Additionally, it has the ability to handle and store test results and give administrators and users feedback.

3) Security: PHP may be used to put security measures in place, like input validation and defense against typical security risks like cross-site scripting (XSS) and SQL injection.

4) Front-end integration: PHP is capable of managing requests from the user interface, interpreting input from the user and sending back messages as required. In order to obtain or save data, it can also manage communication with the database.

III. DATABASE DESIGN

1) Users: Stand in for people who take tests using the system. The Roles entity defines the roles that each user can have, such as administrator, instructor, or student.

2) Tests: Showcase the tests that are accessible via the website. Every test can be categorized into one or more groups

3) Questions: List the questions that are included in every exam. The replies entity represents the several possible replies for each question.

Outcomes: Show the outcomes of every test attempt made by each user. Every result is associated with a particular person and test, and it may contain information about the user, including score and attempt date.

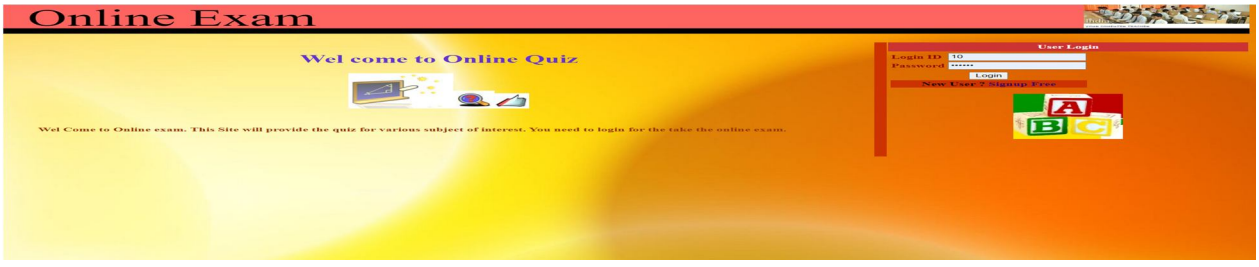
4) Subjects or categories to which tests can be assigned are represented by the categories. There may be more than one test assigned to each category.

5) Test Questions: Indicates which questions are part of each test and shows the relationship between tests and questions.

The essential entities and relationships in a website for online test-taking can be modeled using the fundamental framework provided by this ER diagram. Additional entities and relationships can be required, depending on the particular requirements of the system.

IV. WEBSITE FUNCTONALITY

A. User authentication (login, registration, password recovery)



©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

1) Registration Page: Make a register.php form that asks users to provide their email address, password, and username.
2) Registration Logic: After validating the inputs, process the form submission in register.php and insert the user info into the database. Create a login form (login.php) so that people may input their password and username/email.
3) Login Logic: Verify the user's credentials against the database in login.php. Make a session for the user to keep them logged in if the credentials are correct.
4) Password Recovery Page: Make a password recovery form (recover_password.php) so that users can get a link to reset their password by entering their email address.
5) Password Reset Page: After clicking the link in the email, users can input their new password in the password reset form (reset_password.php).
6) Password Reset Logic: Update the user's password in the database and validate the token from the URL in reset_password.php.
7) Session Management: To maintain track of users who are logged in between pages, utilize PHP session management.

Recall to protect user data by employing secure procedures, such as hashing passwords with PHP's password_hash() function. Furthermore, validate inputs to thwart various kinds of attacks and sanitize inputs to stop SQL injection.

```
CREATE TABLE users (
    id INT(11) AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    registration_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

B. Test creation (adding, editing, deleting questions)

Database Structure: You may have tables such as users, tests, questions, and responses if your database is a MySQL database. Add Question Form (HTML): Use HTML to create a form that allows users to enter questions, choose which test they apply to, and add responses. Add fields for the test ID, question content, question type, and answer choices.

PHP Reverse:

Including a query:

Get PHP form data.

Place the query in the questions table and associate it with the appropriate test ID.

Add response choices to the table of answers.

Modifying an Inquiry:

Using the question ID, retrieve the query and its responses from the database.

Present the query and responses in an editable format.

Based on the user's edits, update the questions and answers in the database.

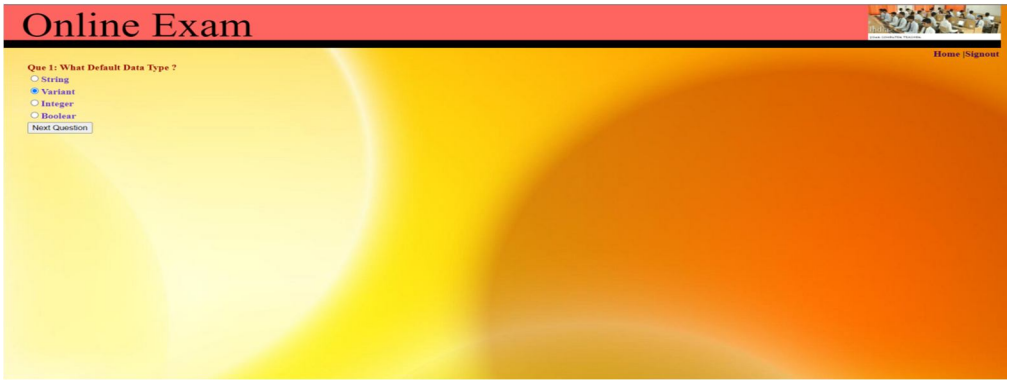
Eliminating a query:

Obtain a request to remove a query.

Remove the query from the database along with any related responses.

Security: To avoid SQL injection attacks, make sure that every user input is cleaned. Passwords are hashed before being entered into a database.

©IJRASET: All Rights are Reserved



C. Test result generation (scoring, feedback)

1) Scoring Procedure: Following the completion of an exam, compute the score using the user's responses and the accurate responses that are kept in the database.

2) Keeping Test Results: Add the user's test result to the test_results table, along with their user ID, test ID, and score. Feedback Submission: Permit test takers to provide feedback. Put the feedback into the feedback table, along with the user ID, test ID, and feedback content.

3) Results Display: Provide users access to a page where they can see the outcomes of their tests and any comments they've left. Security Considerations: Make sure that users cannot change the scoring logic and that it is secure. In order to stop SQL injection and other vulnerabilities, sanitize and verify each input.

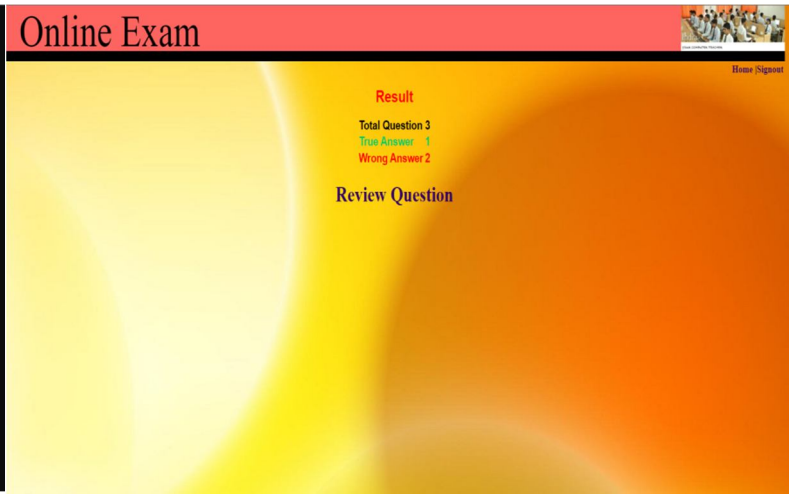
4) User authentication: Make sure that access to test results and feedback submissions are restricted to authorized users only.

5) Data privacy: Adhere to recommended standards for data privacy and make sure that private data, including test results, is secure and only available to those who are permitted.

You may build an online test-taking PHP website with features for scoring, submitting feedback, and displaying results by following these instructions.

```
CREATE TABLE test_results (
    id INT(11) AUTO_INCREMENT PRIMARY KEY,
    user_id INT(11) NOT NULL,
    test_id INT(11) NOT NULL,
    score INT(11),
    date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (test_id) REFERENCES tests(id)
);

CREATE TABLE feedback (
    id INT(11) AUTO_INCREMENT PRIMARY KEY,
    user_id INT(11) NOT NULL,
    test_id INT(11) NOT NULL,
    feedback_text TEXT,
    date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (test_id) REFERENCES tests(id)
);
```



V. IMPLEMETATION

A. Description of the website's features and functionalities

1) User registration and authentication: In order to access the website's features and exams, users must first register an account and log in.

2) Exam Selection: Candidates can select the exam they wish to take, including interviews, the GATE, the CAT, and practice exams for several programming languages.

3) Mock Exams: The website offers a collection of practice exams with question kinds, difficulty levels, and time constraints that mimic the real exam format.

4) Question Bank: A collection of questions that address a range of subjects related to the chosen test, giving users the opportunity to efficiently hone their skills and get ready.
5) Test Personalization: To concentrate their study, users can designate particular subjects, degrees of difficulty, and kinds of questions for their mock exams.
6) Real-time Feedback: Instant feedback to assist users comprehend and learn, including explanations and right responses for answers.
7) Performance analytics: A thorough examination of test results, including strengths, weaknesses, and scores, to monitor development and pinpoint areas in need of improvement.
8) Time management tools: Tools like timers and pacing guidelines that assist users in efficiently managing their time during exams.
   Discussion Forums: Online discussion forums where users can exchange advice, discuss exam-related subjects, and look for advice from professionals and peers.
9) Mobile Compatibility: The ability to take tests while on the go through compatibility with mobile devices. Implementing security measures is necessary to stop cheating and guarantee the validity of the tests.

The overall goal of an online test-taking website is to give customers a comprehensive platform so they may confidently and successfully prepare for their tests, including computer language, CAT, GATE, and interview mock exams.

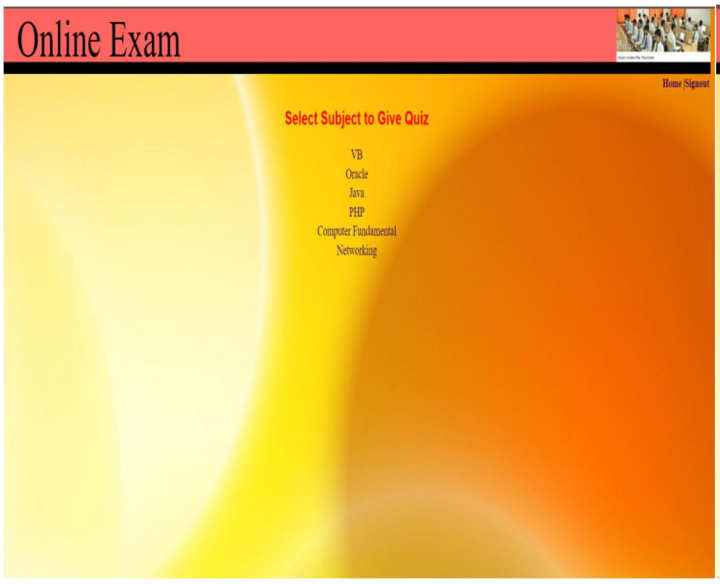
VI. RESULT

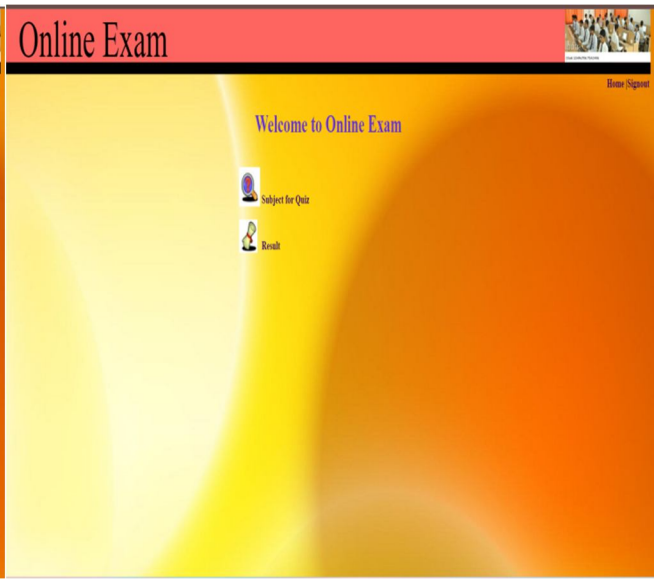
A. *Analysis of test data (average scores, question difficulty)*

You can use the following actions to evaluate test data on a PHP-built online test-taking website, including average scores and question difficulty:

1) Gathering Information: After tests are finished, enter user replies, right answers, and question levels into your database. Make certain that every question has a corresponding level of difficulty.
2) Compute Average Scores: To determine the average scores for every exam, use SQL queries. By dividing the total number of tests taken by the sum of all test scores, you may find the average score.
3) Determine Question Difficulty: The percentage of users who correctly answered the question can be used to determine the difficulty of the question. The difficulty decreases as the proportion increases.
4) Results Are Shown: On the dashboard or analytics page of the website, provide the average scores and question difficulty levels in an easily comprehensible way (e.g., tables or charts).
5) Data Updating: To make sure the information is accurate and up to date, update the analysis on a regular basis using new test data.

You may enhance the efficacy of your online test-taking website by gaining insights into user performance and question difficulty through the analysis of test data.





VII. CONCLUSION

A. *Summary of key findings*

1) Average Scores: Users' overall performance can be inferred from the average scores they receive on various tests. Lower average scores can imply more difficult questions, while higher average scores might indicate that the test's material is generally simple.
2) Question Difficulty: The degree of difficulty of a question can be ascertained by examining the proportion of users who correctly answered each one. It is likely that questions with a high percentage of right responses are simpler and those with a low percentage are trickier.
3) User Performance: You may determine which users performed especially well or poorly by comparing their individual scores to the average scores. This data can be used to customize study guides or other interventions for users who might require more assistance.
4) Effectiveness of the inquiry: Examining user feedback can also be used to gauge how well a particular inquiry works. It could be necessary to examine or amend questions that a significant portion of users routinely provide inaccurate answers for.
5) Trends Over Time: Examine how user performance and test efficacy have changed over time by monitoring average scores and question complexity. For instance, a decline in average results over time can be a sign that the test's content is getting obsolete or too simple for students to understand.
6) Test Design: The analysis's conclusions can potentially influence test designs in the future. To guarantee that the test appropriately evaluates user knowledge and skills, questions that are frequently answered correctly by the majority of users, for instance, might need to be swapped out for harder ones.
7) Feedback and Enhancement: Lastly, the analysis can assist in pinpointing areas where the online test-taking platform needs to be improved. To improve learning results and user experience, this could involve making adjustments to the test's content, scoring methodology, or user interface.

In general, test data analysis can yield insightful information about test design, question efficacy, and user performance, which can ultimately result in an improved and more enjoyable online testing environment.

B. *Call To Action For Further Research And Development In Online Testing Technologies*

1) Enhanced Security Measures: Create and put into place sophisticated security measures, like biometric authentication, secure browser settings, and anti-cheating systems, to guarantee the integrity and privacy of online exams.
2) Examine and improve adaptive testing algorithms to provide a more effective and individualized testing experience. These algorithms dynamically modify the difficulty of questions based on the user's performance.
3) Utilize machine learning and data analytics methods to examine vast amounts of test data, spot trends and patterns, and produce new ideas to enhance test efficacy and design.
4) Create cutting-edge remote proctoring systems that employ AI and machine learning to keep an eye on test takers and flag questionable conduct.
5) Accessibility and Inclusivity: Make sure that people with disabilities may fully participate in online exams by enhancing accessibility features such as screen reader compatibility and other input techniques in online testing platforms.
6) Learning Management System (LMS) Integration: Improved LMS platform integration will speed up the development, administration, and grading of tests, giving teachers and students a smooth experience.
7) User Experience (UX) Design: To improve online testing platforms' usability and make them more intuitive, captivating, and user-friendly, do usability testing and user research.
8) Cross-Platform Compatibility: To enable people to take tests on any device, at any time, anywhere, make sure that online testing platforms are compatible with a variety of browsers and devices.
9) Enhance accessibility capabilities, such as screen reader compatibility and alternative input methods, in online testing platforms to ensure that those with impairments may participate completely in exams. This would promote inclusivity.
10) Better Learning Management System (LMS) platform integration will expedite test creation, administration, and grading while providing instructors and students with an easy-to-use environment.
11) User Experience (UX) Design: Conduct user research and usability testing to enhance the usability of online testing platforms and make them more compelling, intuitive, and user-friendly.

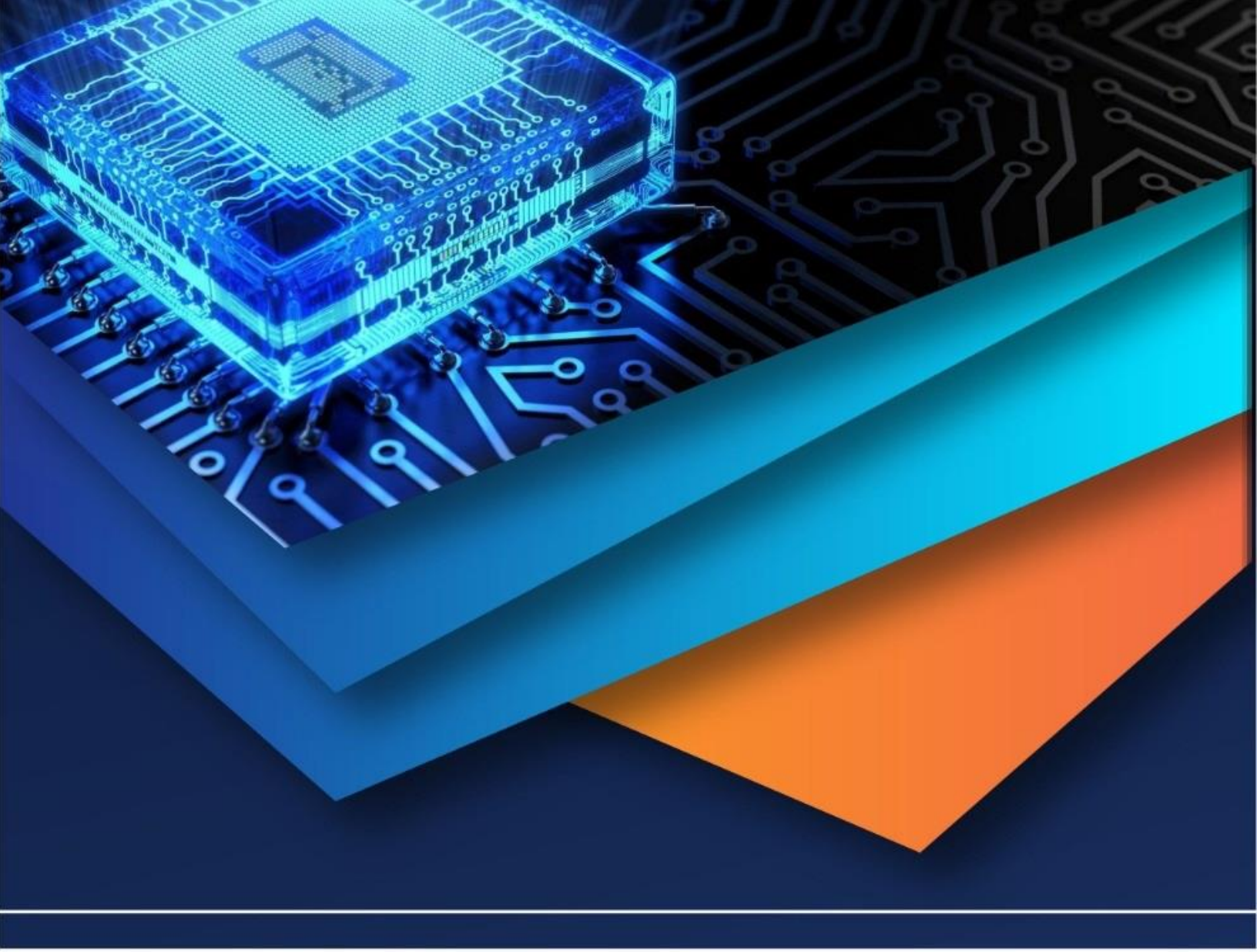
©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

12) Cross-Platform Compatibility: Ensure that online testing systems are interoperable with a range of browsers and devices so that individuals may take tests on any device, at any time, anywhere.

REFERENCES

[1] (2008) "Comparison of Student Performance in Paper-Based Versus Computer- Based Testing," Journal of Education for Business, No. October, pp. 13-18.
[2] Biggs, J. and Tang, C. (2011) Teaching for Quality Learning at University, Berkshire, McGraw Hill, 4th ed.
[3] Chao, K.J., Hung, I.C. and Chen, N.S. (2011) "On the design of online synchronous assessments in a synchronous cyber classroom," Journal of Computer Assisted Learning, Blackwell Publishing Ltd, p. 1-17
[4] Elliott, B. (2008), "Assessment 2.0: Modernising assessment in the age of Web 2.0," Scottish Qualifications Authority. Retrieved May 16, 2012, from http://www.scribd.com/doc/461041/Assessment-20.
[5] Elliott, B. (2009) "E - PEDAGOGY: Does e-learning require a new pedagogy?," Scottish Qualifications Authority. Retrieved May 16, 2012, from http://www.scribd.com/doc/932164/E-Pedagogy. Frankl, G., Schartner, P. and Zebedin, G. (2011) "The 'Secure Exam Environment' for Online Testing at the Alpen-Adria-Universität Klagenfurt/Austria," World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, Hawaii, Association for the Advancement of Computing in Education (AACE), pp. 498-505.
[6] Hewson, C. (2012) "Can online course-based assessment methods be fair and equitable? Relationships between students' preferences and performance within online and offline assessments," Journal of Computer Assisted Learning, Blackwell Publishing Ltd.
[7] Kahneman, D. (2011) "Thinking, Fast and Slow", New York, Farrar, Straus and Giroux. Marriott, P. (2009) "Students' evaluation of the use of online summative assessment on an undergraduate financial accounting module," British Journal of Educational Technology, Blackwell Publishing Ltd, Vol. 40 No. 2, pp. 237-254.
[8] Müller, A. and Schmidt, B. (2009) "Prüfungen als Lernchance": Sinn, Ziele und Formen von Hochschulprüfungen," Zeitschrift für Hochschulentwicklung, Vol. 4 No. 1, pp. 23-45. Müller, F.H. and Bayer, C. (2007) "Prüfungen: Vorbereitung - Durchführung - Bewertung," inHawelka,B.,Hammerl,M. and Gruber,H. (Eds.),Förderung von Kompetenzen in der Hochschullehre, Kröning, Ansager, pp. 223-237

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)