British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 6, Issue 1, 2023 (pp. 66-82)
www.abjournals.org

DESIGN OF QUESTION GENERATOR SYSTEM (QPGS) USING FISHER-YATES SHUFLING ALGORITHM

Umoh Enoima Essien

Department of Computer Science, University of Cross River State

enoimaumoh@unicross.edu.ng

Cite this article:

Umoh Enoima Essien (2023), Design of Question Generator System (QPGS) Using Fisher-Yates Shufling Algorithm. British Journal of Computer, Networking and Information Technology 6(1), 66-82. DOI: 10.52589/BJCNIT-ZCK1BFKB

Manuscript History

Received: 6 Aug 2023

Accepted: 26 Sept 2023

Published: 18 Oct 2023

ABSTRACT: *The process of generating question papers for educational assessments is a crucial task in the field of academia. The traditional manual approach to question paper creation is time-consuming and prone to human error. In response to this challenge, the Question Paper Generation System (QPGS) has emerged as a powerful solution that leverages technology to automate and streamline the question paper generation process. The Question Paper Generation System has a profound impact on the education sector by enhancing the quality and efficiency of assessments. It empowers educators to create assessments that align with educational goals and standards, ultimately benefiting both teachers and students. Moreover, it contributes to the overall modernization of educational practices by integrating technology seamlessly into the assessment process. This paper provides an overview of the Question Paper Generation System, highlighting its key features, benefits, and impact on educational institutions. QPGS combines advanced algorithms, database management, and user-friendly interfaces to efficiently produce high-quality question papers tailored to specific subjects, courses, and assessment levels. For efficient randomization and shuffling of the questions in the question bank, the Fisher-Yates shuffle algorithm, also known as the Knuth shuffle or the Durstenfeld shuffle, was used. A hybrid of the Object-Oriented Analysis and Design (OOAD) together with the Structured System Analysis Methodology (SSADM) with Unified Modeling Language (UML) was adopted for the design of the system. The system was implemented using HTML, CSS and JavaScript as the front end, while the back end which deals with the system's logic was implemented using PHP and MySQL. The results obtained were tested using several test strategies.*

KEYWORDS: Question Paper Generation, Fisher-Yates shuffle algorithm, Unified Modeling Language.

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 6, Issue 1, 2023 (pp. 66-82)

www.abjournals.org

INTRODUCTION

Technological advancement has caused a great shift and brought about a lot of changes in every segment of the global economy, with its glaring and evident benefits, thus rendering previous methodologies obsolete. It is aimed at increasing productivity and manufacturing prowess, reducing cost and labor, and eliminating human error (Fenil et al., 2017). Also, the educational sector is not left out either. Automation such as Robotic Process Automation (RPA) and Artificial Intelligence (AI) deliver a lot of benefits to trainers, students and academic instructors. Some of those potential tasks for automation in the educational sector include those niggling but vigorous automations, such as scheduling, keeping track of attendance, assignment grading, and question paper generation (Shajahan, 2017). Most of the times, students get only recall (bottom of the Bloom's Taxonomy) type questions during assessment phase but professionally they are assessed on all aspects as per blooms taxonomy. This creates a knowledge gap which is not revealed in the former part. Our present education system does not reflect on this concern and intellectual growth of students' founders. Teachers' goal is to pass on knowledge and enhance students' learning ability. All courses have LO's (Learning Objectives); assessment for such courses is verification by the course's teacher to check whether the students have met or achieved the objectives or not. But designing such questions to test learning (categorized by Bloom) of the students with respect to particular courses needs enormous experience and is a time consuming activity. This would be difficult for budding teachers and even experienced teachers may not be able to create such questions frequently as there are several tests, quizzes and exams in a year in any academic house. The complex curriculum puts pressure on the question paper designer (QPD) to assess students along with covering all the objectives of the course. So, there is a need to develop an intelligent technology-based system for generating standard question paper that examines the overall growth of examinee. A standard question paper contains eclectic questions from a rich question bank (database) according to the organization's course curriculum (Purohit et al., 2012).

At the end of every academic year for every examination conducted in any academic course, teachers need to create a variation of question papers. It might be difficult for the teachers to cover the different features of the course outcomes and eventually evade duplication of questions in the forthcoming examinations. No systematic procedure is provided for that; therefore, the quality of the question paper will depend solely on an individual teacher's proficiency and experience most times, and this may degrade the quality of the question paper. According to research, a question paper is of good quality if it is a combination of questions which are supervised by different criteria such as difficulty level, distribution of marks across the question paper in form of paper pattern and the examination type. The procedure involved in composition of an equitable examination paper manually is challenging and complex. The standard of the examination paper relies on diverse sets of specifications; hence, taking into account the distinct levels of learners is also a crucial parameter and the course outcomes also play a vital role in planning a systematic question paper. So associating the learning outcome of the course to the examination paper is also a great job (Fenil, Hital & Ashwir, 2017). With the profound dispersal of technology in the area of education, acquiring technology to smoothen the technique of examination paper creation is a pure option and creation of an extensively vast question bank and automatic exam paper generation furnishes a key provision to the issue encountered during the manual composition of examination papers. Automatic creation of examination paper yields a stage to create a well-organized examination paper and also the automation would smoothen the incorporating of many elements that determine the

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

quality of a question paper. With respect to this, we introduce an Automatic Examination Question Paper Generator System.

This system will reduce the different bottlenecks associated with the generation of question papers and also serve as a repository for storing and preserving questions for future references. The aim of this paper is to design and develop an automated system for generating question papers using the *Fisher-Yates* shuffle, enabling administrators to input a collection of questions along with their respective weights. The Fisher-Yates shuffle is an algorithm employed to achieve a flawless shuffling of items using a random number generator. The name of the algorithm is derived from Ronald Fisher and Frank Yates, who initially introduced this method in their book in 1938. Subsequently, Donal Knuth and Richard Durstenfeld introduced an enhanced version of the algorithm in 1964. Differing from the approach of swapping items at distinct indices, this algorithm generates a random number 'k' within the range of elements within an array. With each iteration, the algorithm modifies the last element within the range. This characteristic ensures that the random generator operates within a new range in every iteration, thereby producing a distinct number on each occasion.

REVIEW OF RELATED LITERATURE

Several researchers have explored the field of Examinations Question Generation, and the following section provides an overview of prior studies: In a study by Fenil, Hital, and Ashwini (2017), an Automated Question Paper Generator System was created specifically for use in autonomous educational institutions in India. This system was built using the Java programming language. The system, which was fully operational, included databases of courses, questions, and question paper formats. The process involved applying an algorithm to the stored question pool, resulting in the generation of a PDF-format question paper. The algorithm employed by their system utilized randomization, as outlined below.

For N number questions accessible in the question bank database, the following are the steps:

1. Generate a list ‘L’ number of N elements
2. Generate a random number ‘n’ such that 1<=n
3. If n is an element of L, then go to Step 2, else reserve n in the List L
4. Choose a question from question bank database corresponding to n, whose flag == true
5. For the question, set flag == false.

Tendolkar et al. (2017) investigated the challenges and obstacles linked to the manual creation of question papers by educators, proposing a remedy in the form of an Automatic Question Paper Generating System. Developed with the ASP.Net programming language, this system generated question papers using an existing question repository in the database. The process involved the utilization of fuzzy logic and the apriori algorithm. The research outcome yielded a functional web application that underwent testing and demonstrated successful operation.

Mohandas et al. (2015) formulated an Automated Question Paper Generator System using Microsoft Visual Studio and SQL Server for the construction of the question bank. The system

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

harnessed the capabilities of the Fuzzy logic algorithm, allowing for an impartial and randomized selection of questions from the question bank.

Naik et al. (2014) introduced an Automatic Question Generation System comprising various modules such as user administration, subject selection, difficulty level specification, question entry, question management, paper generation, and paper management. Through a meticulous design process, the system adeptly scrutinized and composed examination papers with a high success rate. Their approach employed a shuffling algorithm for randomization purposes. Within this algorithm, users specified the subject, question type, and difficulty level, and the system automatically generated the examination paper accordingly. The editing of questions was facilitated through a Word processor, with the final paper being storable in ".doc" file format. Noteworthy attributes of the system included user-friendly operation, a well-designed interface, optimal usability, robust security, and high stability, along with reliability.

Alldabe et al. (2006) introduced ArikIturri, an Automatic Question Generator designed for crafting Basque language test questions. This generator relies on linguistically analyzed real corpora, represented in Extensible Markup Language (XML), as its information source. ArikIturri employs Natural Language Processing (NLP) tools and operates with a data bank containing morphologically and syntactically analyzed sentences, identifying phrase chunks. Input for ArikIturri is given in XML format, while the generated outputs are instances of questions defined using XML markup language. The system automatically generates question instances, utilizing both NLP tools and specialized linguistic information for question formulation.

Sung, Lin, and Chen (2006) outlined an approach for creating an Automatic Quiz Generation system within a global English E-learning context. This system evaluates learners' understanding of text content and English skills based on a provided English text. The auto-quiz procedure involves parsing the English text into a semantic network representation and progressively enriching this network with intrinsic knowledge like English grammar and writing styles, as well as extrinsic knowledge from sources like WordNet and statistics from corpora or search engines. The quiz generation process then generates quizzes from the text, considering the learner's comprehension skills and their learning status and needs, such as proficiency and common errors. A prototype system was developed using Java programming language, exhibiting high efficiency.

Purohit et al. (2012) shed light on the limitations of various computer-based Question Paper Generators, highlighting issues such as inadequate databases for question recall and the absence of historical records for previously generated question modules. They proposed an adaptive question bank management system as a solution, intelligently selecting questions from a rich question bank and representing the question model based on inputs from the question paper designer (QPD). This system incorporates a concept map developed using a graphical method, utilizing hierarchical domain knowledge. The concept map, integrated with the question bank, ensures question modeling based on criteria like Bloom's Taxonomy, difficulty level, and marking scheme. The evaluation of the generated question model provides feedback on students' overall understanding, serving as an advantage for institutions aiming to enhance their students' growth.

Liu and Wang (2013) suggested that it is possible to create examination papers through the utilization of algorithms such as the random variable algorithm, backtracking algorithm, and

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

artificial algorithm. Nonetheless, this paper proposes that the Fisher-Yates Shuffle algorithm stands as the optimal choice for generating exam papers, as it addresses the time processing limitations that arise with the random variable algorithm and backtracking algorithm, especially when dealing with extensive question repositories. The Fisher-Yates Shuffle algorithm has found application in various methods and case studies that involve the utilization of the randomization function. This algorithm offers the advantage of efficient randomization and boasts an optimal algorithmic complexity, likely in the order of O(n) (Saeed et al., 2014). Furthermore, its widespread use stems from its ability to ensure non-repetitive and non-duplicated randomization of resulting data [5]. Another notable benefit of the Fisher-Yates algorithm is its capability to generate unbiased outcomes, where each permutation of an array possesses equal probabilities. However, it is worth noting that this algorithm also exhibits limitations, specifically in terms of the sorting process, which follows a sequential sorting approach (Sharmila et al., 2016).

SYSTEM DESIGN AND METHODOLOGY

System Methodology

In this research work, an extensive study of the design of question paper generation was conducted with a special focus on Shuffling algorithm using enhanced Fisher Yates Shuffle Algorithm. The design model adopted in this work is a hybrid of the Object-Oriented Analysis and Design (OOAD) together with the Structured System Analysis Methodology (SSADM) with Unified Modeling Language (UML). UML is a graphical language that allows people who design software systems to use an industry standard notation to represent them. UML is now the most widely used graphical representation scheme for modeling object systems. An attractive feature of UML is its flexibility. UML modelers are free to use various processes in designing systems. UML is a complex, feature-rich graphical language (Ele & Adesola, 2013).

System Design

System Functional Requirements

The specific functionalities offered by the question paper generation system include:

i. It has the ability to select subject(s) and topic(s).

ii. It has option to specify difficulty levels.

iii. It contains different question types, such as multiple choice, short answer, essay, etc.

iv. It possesses the ability for randomization of questions or answer choices.

v. The system can preview the generated question paper.

vi. It is flexible and can add custom questions.

vii. Supports images, diagrams, and equations.

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

Non-Functional Requirements

The non-functional aspects of the system include:

i. Performance: The system can support more than 10 users simultaneously.

ii. Scalability: The system has the ability to accommodate a growing question database.

iii. Usability: The system possesses an intuitive user interface (UI) for teachers with varying technical skills.

iv. Reliability: It has the ability to minimize downtime during question paper creation.

v. Maintenance: Updating of question database and system components is easy.

Algorithm for the Question Generator

Step 1: Start

Step 2: Declare variables total_quest, sub_quest and total_array[], total_marks, I, j, k, rand, mark, final_mark, l.

Step 3: Read values total_quest, sub_quest, total_mark.

Step 4: Set i = 0, j = 0, k = 0, l = 0.

Step 5: Repeat these steps until i>(total_quest-1)

5.1: Declare variable sub_arr[]

5.2: Repeat these steps until j > (total_quest – 1)

5.2.1: Set rand = Generated random number

5.2.2: Repeat these steps until k > (total_quest – 1)

5.2.2.1: if rand = total_arr[k]

Set rand = Regenerated random number

Else

Set sub_arr[j] = rand.

Step 6: Repeat these steps until L>(total_quest – 1)

6.1: Select marks of question from question bank using elements of the total_arr as id

6.2: Set mark = selected mark.

6.3: final_mark = final_mark + mark.

6.4: if final_mark = total_marks

Go to step 7

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

Else

Repeat step 5

Step 7: Select questions from question bank using the total_arr[] elements as id

Step 8: Print questions

Step 9: Stop

Figure 1 illustrates the question paper generation system in stepwise order, while Figure 2 is the flowchart of a typical Fisher-Yates Shuffle algorithm as also deployed by Wijaya and Chang (3023).

.

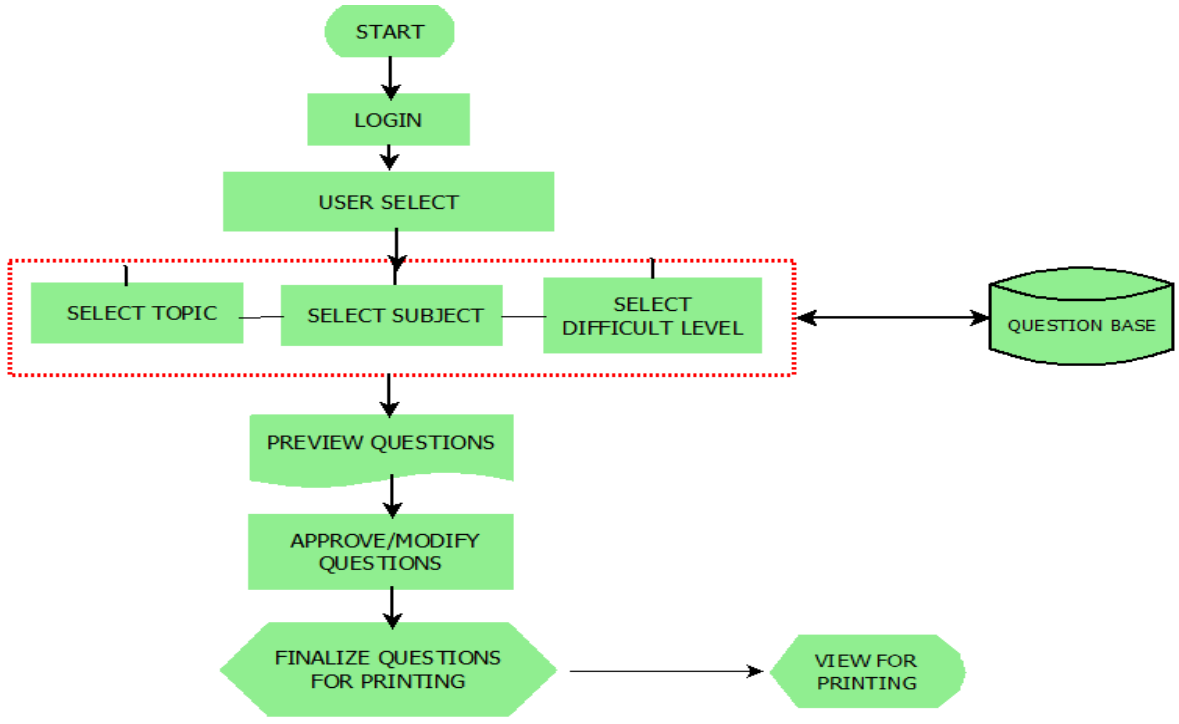


Figure 1: Stepwise Order of the Question Generator System

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 6, Issue 1, 2023 (pp. 66-82)
www.abjournals.org

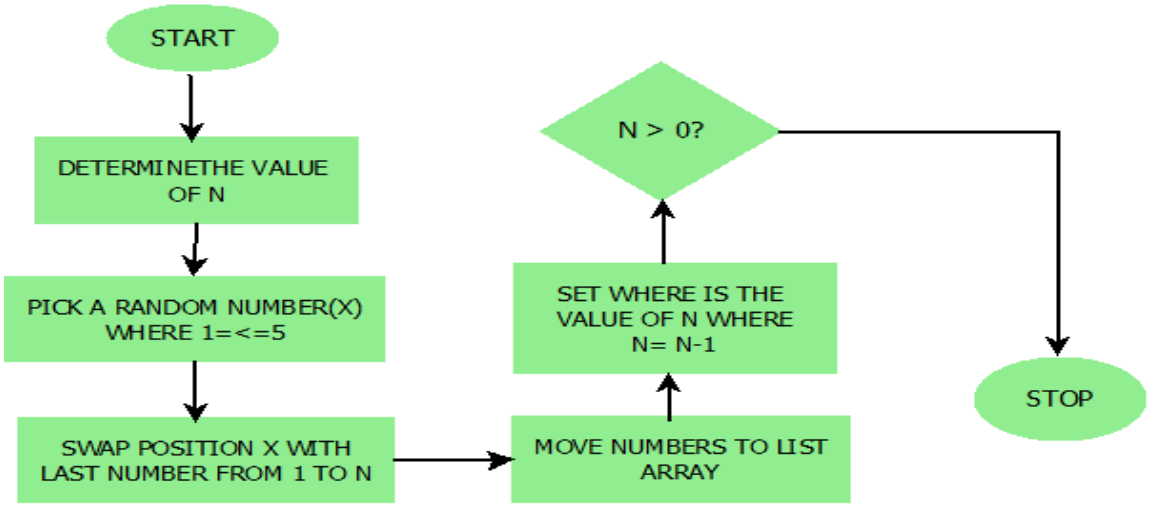


Figure 2: Flowchart of a Typical Fisher-Yates Shuffle Algorithm for Question Generator System (Wijaya & Chang, 3023)

Use Case Design

A Use Case outlines how a system or its component operates, presenting a depiction of various sequences of actions, including alternate versions, that the system executes to produce a noticeable outcome of significance to an external entity, which could be a person, process, or object, interacting with the system, subsystem, or category (Ele, Ofem & Obono, 2018). A Use Case diagram of the question paper generator system is represented in Figure 3.

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 6, Issue 1, 2023 (pp. 66-82)
www.abjournals.org

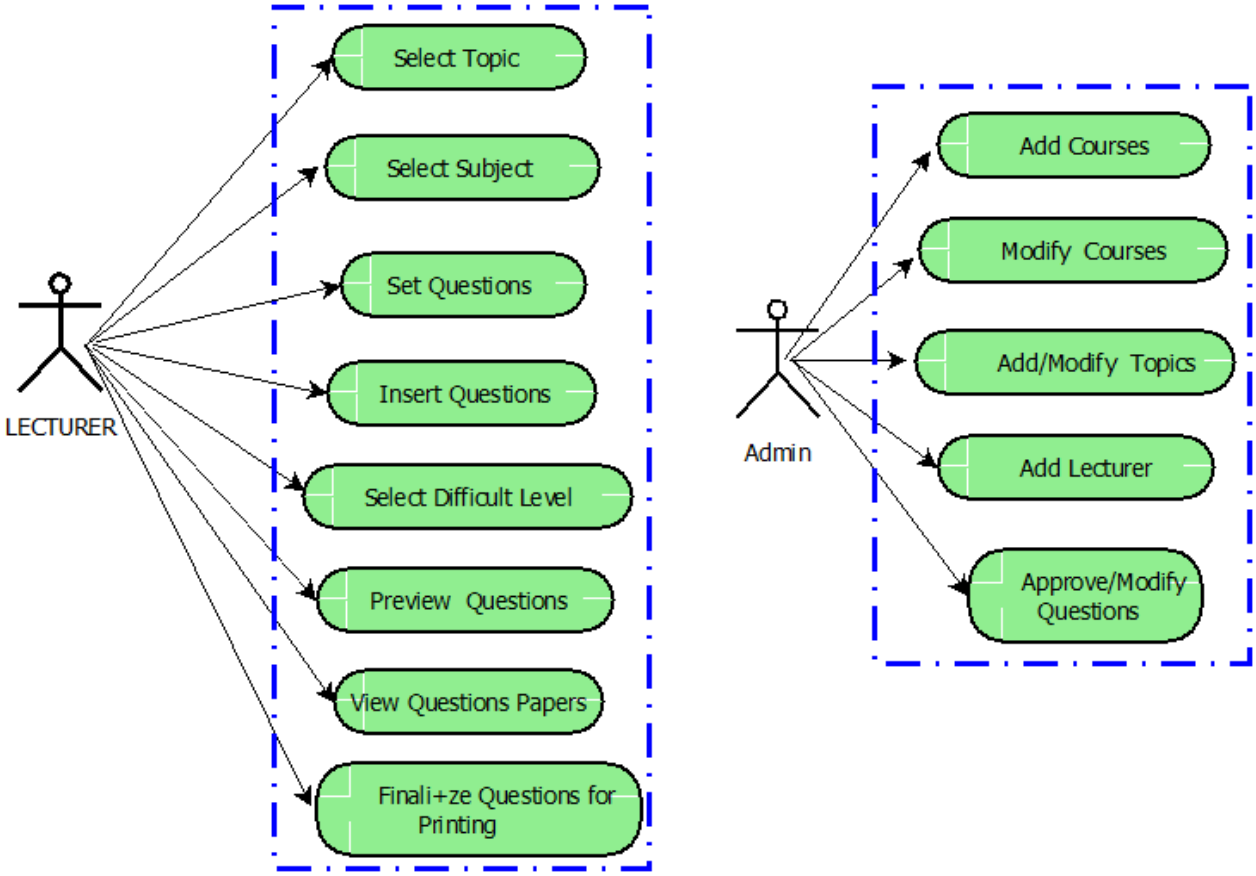


Figure 3: Use Case Diagram Depicting the Various Sequences of Actions in the Question Generator System

Class Diagram

The class diagram establishes the system design's abstraction at an architectural level. It enhances the organization of attributes or data and the operations or functions of the question paper generation system into a unified and separate entity (Ele**,** Umoh, & Obono, 2021). The class diagram for the model system is displayed in Figure 4.

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 6, Issue 1, 2023 (pp. 66-82)
www.abjournals.org

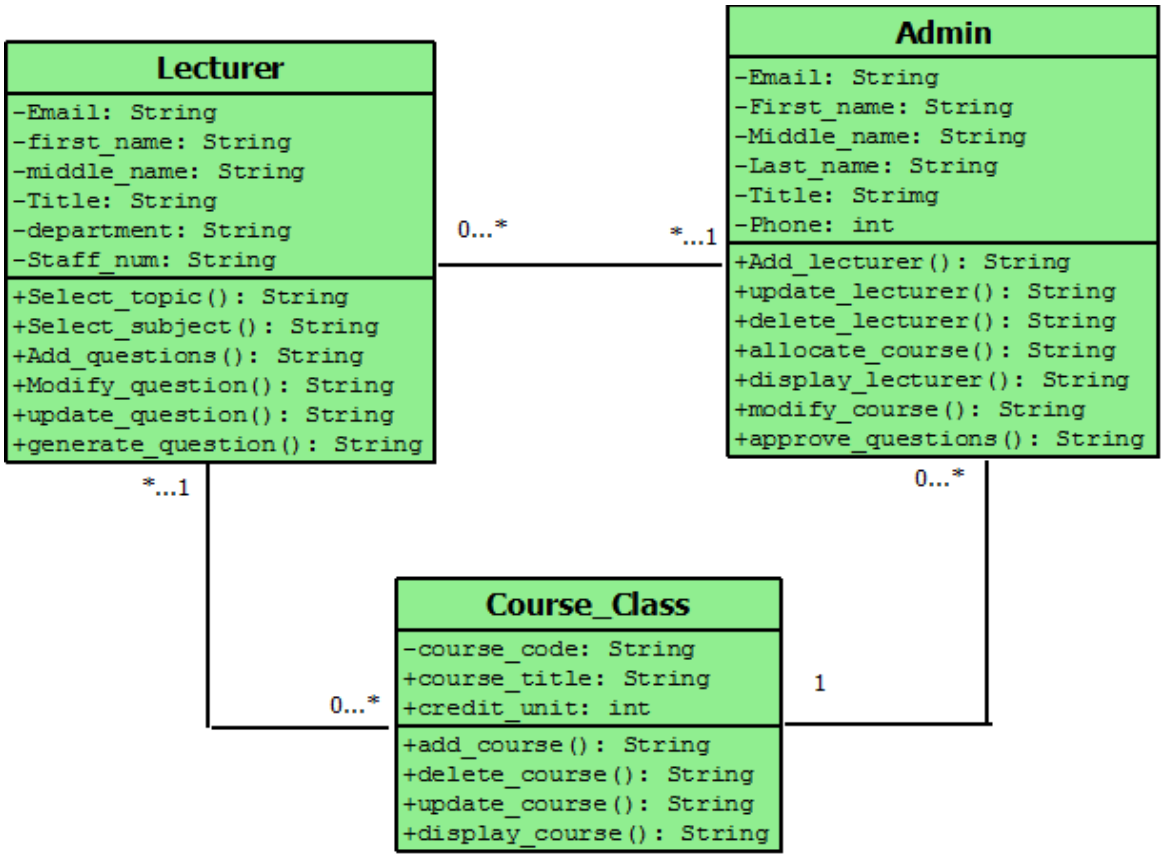


Figure 4: Class Diagram for the Model System for the Question Paper Generation.

Database Design

Table Design

Database table design refers to the process of planning and creating the structure, layout, and attributes of a table within a relational database management system (RDBMS). The goal of database table design is to ensure efficient data storage, retrieval, and manipulation while maintaining data integrity and minimizing redundancy. Proper table design is crucial for organizing data effectively and supporting the desired functionalities of the database system.

Table 3.1: Lecturer Table

Field	Data Type	Foreign	Null
Email	Varchar(191)	Yes	Not Null
Password	Varchar(191)	No	Not Null
First_name	Varchar(191)	No	Not Null
Middle_name	Varchar(191)	No	Not Null
Last_name	Varchar(191)	No	Not Null
Title	Varchar(191)	No	Not Null
Phone	Varchar(191)	No	Not Null

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

Table 3.2: Lecturer_course Table

Field	Data Type	Foreign	Null
Email	Varchar(191)	Yes	Not Null
Course_code	Int(11)	Yes	Not Null
Session	Varchar(191)	No	Not Null

Table 3.3: Course Table

Field	Data Type	Primary	Null
Course_code	Int(11)	Yes	Not Null
Course_title	Varchar(191)	No	Not Null
Credit_hour	Int(11)	No	Not Null

Table 3.4: Question_paper Table

Field	Data Type	Foreign	Null
Course_code	Int(11)	Yes	Not Null
Session	Varchar(191)	No	Not Null

Table 3.5: Question Table

Field	Data Type	Foreign	Null
Course_code	Int(11)	Yes	Not Null
Marks	Int(11)	No	Not Null
Difficulty	Varchr(191)	No	Not Null
Pattern	Varchar(191)	No	Not Null
Question	Text	No	Not Null

Entity Relationship Diagram (ERD)

An Entity Relationship Diagram (ERD) is a visual representation that depicts the relationships between different entities in a database. It is a powerful tool for designing and illustrating the structure of a database system. Figure 5 is a representation of the ERD for the question paper generation system.

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 6, Issue 1, 2023 (pp. 66-82)
www.abjournals.org

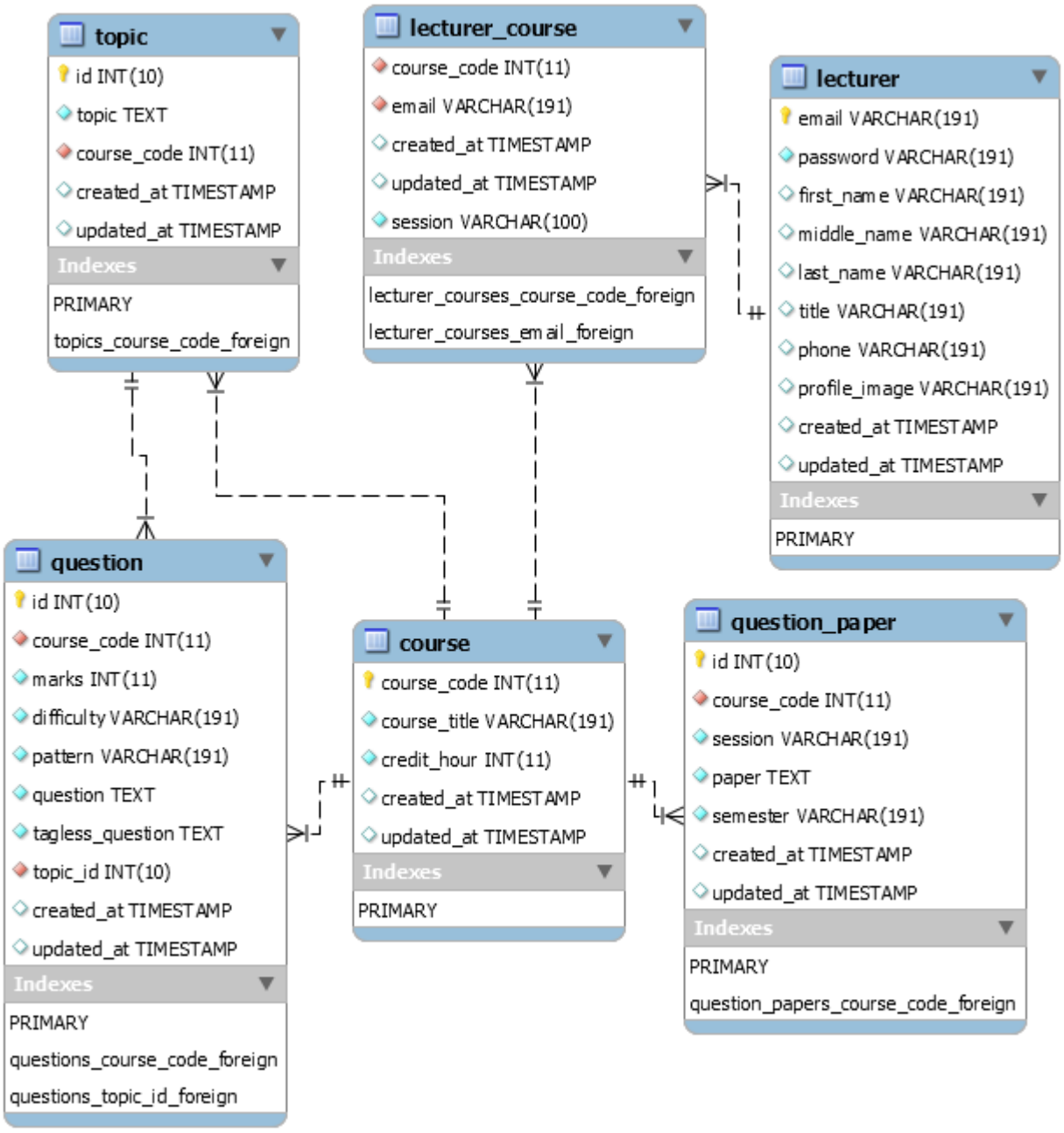


Figure 5: Entity Relationship Diagram (ERD) for the Question Generator System

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 6, Issue 1, 2023 (pp. 66-82)
www.abjournals.org

IMPLEMENTATION AND RESULTS

The system was developed and implemented based on the requirements gathered during the requirements analysis phase from interviews, planning, design, etc. The implementation consists of the front end and back end. The front end, which is the user interface, was developed using HTML, CSS and JavaScript while the back end, which deals with the system's logic, was developed using PHP and MySQL. The research paper's system underwent testing through a range of test approaches. The research work involved performing manual unit testing, where each component of the system was individually created as an independent module and then rigorously tested to verify their compatibility and functionality. Figures 6 represents the interfaces of the system at various modules.

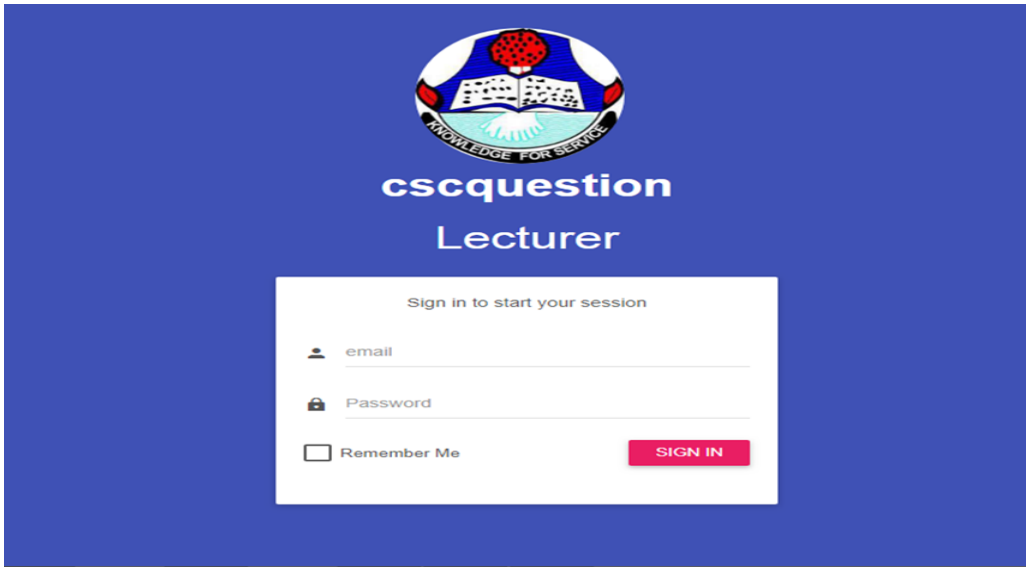


Figure 6: Form for Inserting Questions into the System

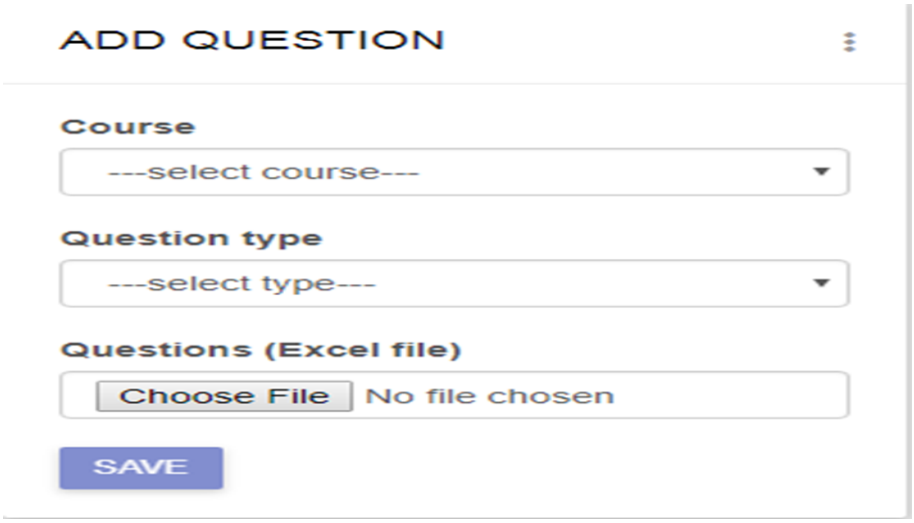


Figure 7: Courses That a Lecturer Has Been Allocated

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 6, Issue 1, 2023 (pp. 66-82)
www.abjournals.org

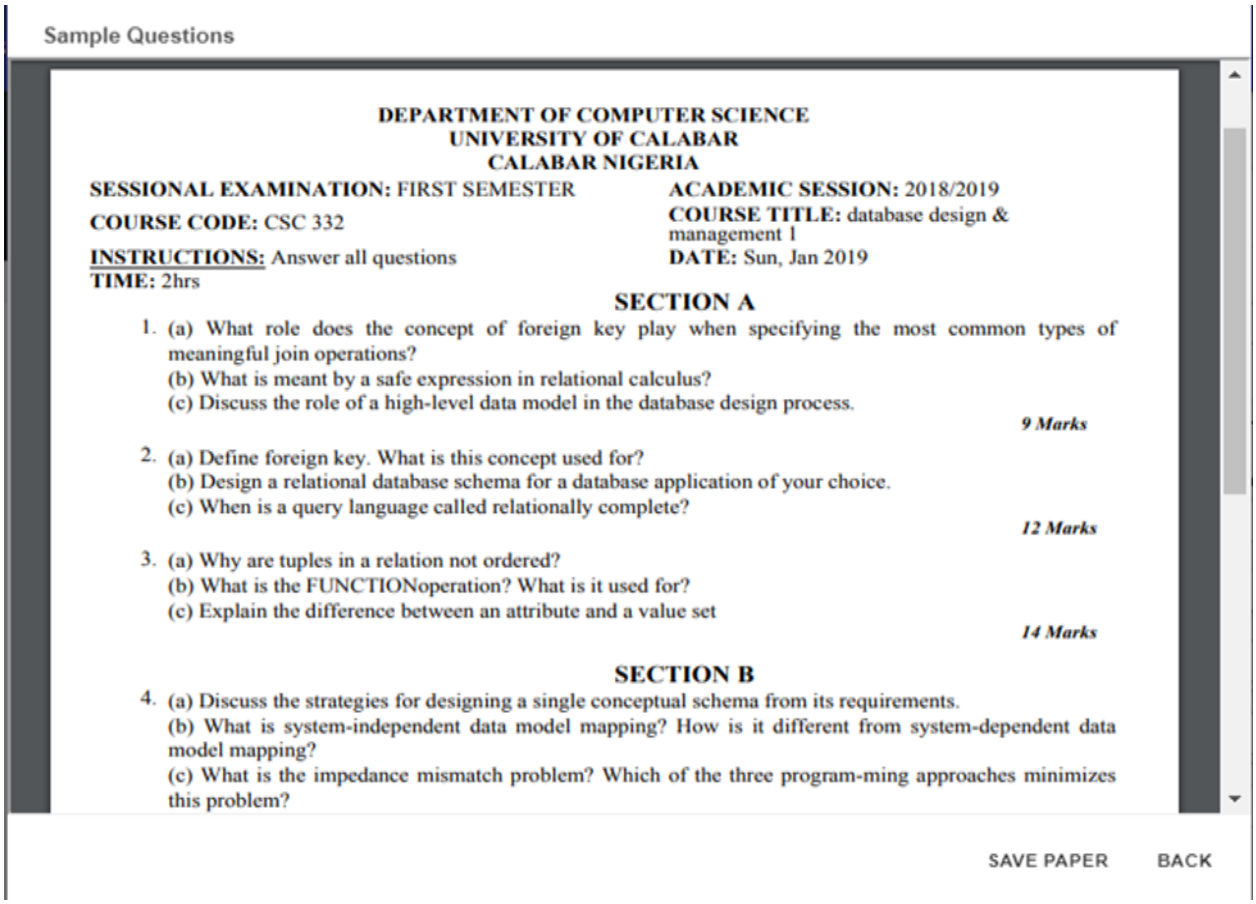


Figure 8: Sample Questions on Preview

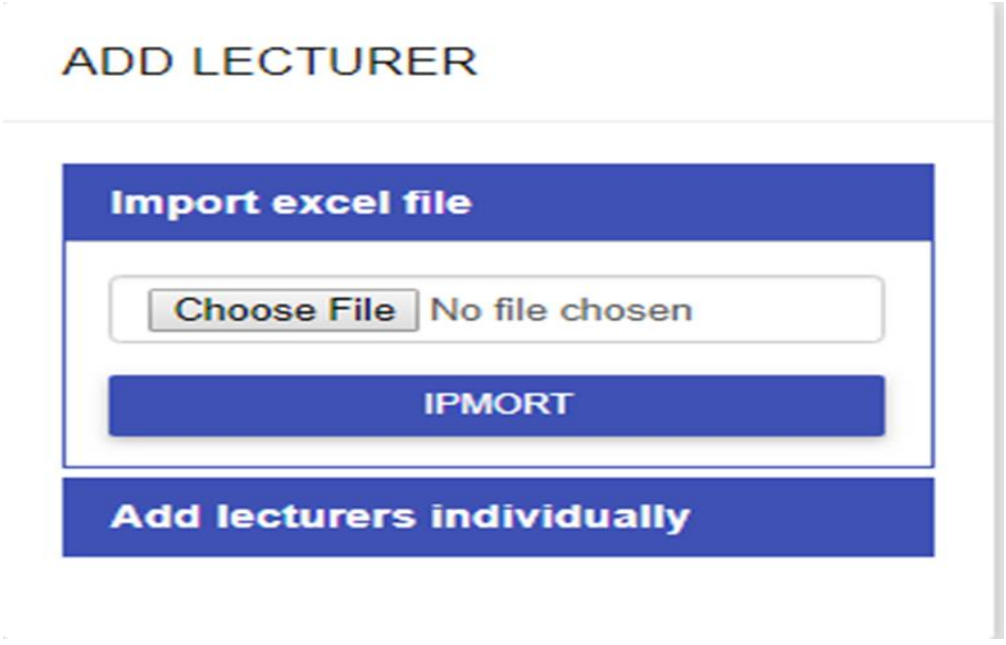


Figure 9: List of Lecturers Available in the System

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 6, Issue 1, 2023 (pp. 66-82)
www.abjournals.org

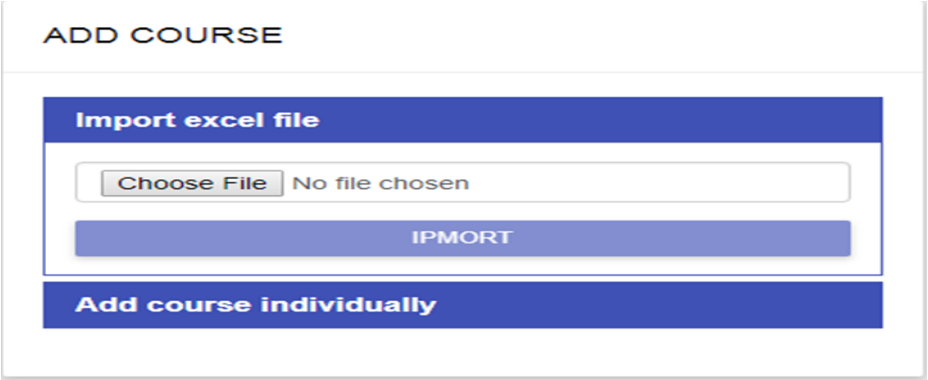
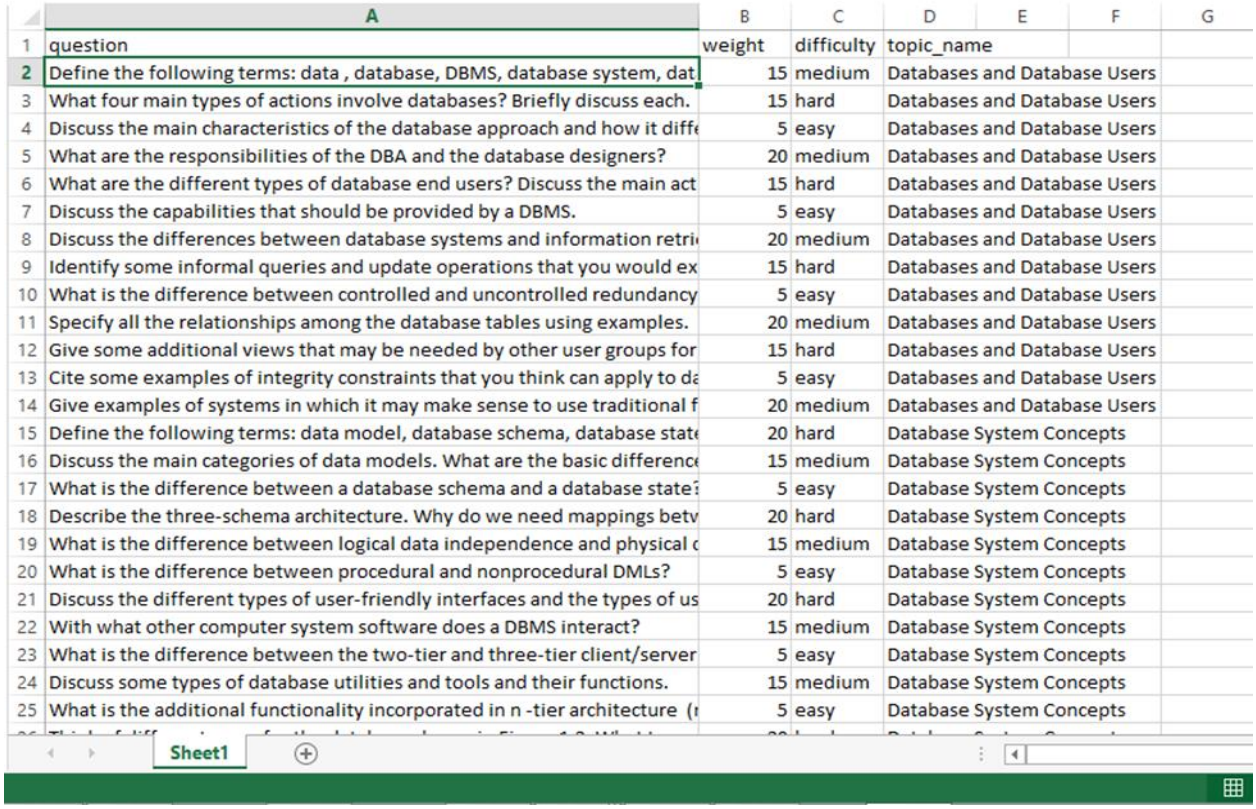


Figure 10: Form for Adding Topics for Courses



	question	weight	difficulty	topic_name
2	Define the following terms: data , database, DBMS, database system, dat	15	medium	Databases and Database Users
3	What four main types of actions involve databases? Briefly discuss each.	15	hard	Databases and Database Users
4	Discuss the main characteristics of the database approach and how it diffe	5	easy	Databases and Database Users
5	What are the responsibilities of the DBA and the database designers?	20	medium	Databases and Database Users
6	What are the different types of database end users? Discuss the main act	15	hard	Databases and Database Users
7	Discuss the capabilities that should be provided by a DBMS.	5	easy	Databases and Database Users
8	Discuss the differences between database systems and information retri	20	medium	Databases and Database Users
9	Identify some informal queries and update operations that you would ex	15	hard	Databases and Database Users
10	What is the difference between controlled and uncontrolled redundancy	5	easy	Databases and Database Users
11	Specify all the relationships among the database tables using examples.	20	medium	Databases and Database Users
12	Give some additional views that may be needed by other user groups for	15	hard	Databases and Database Users
13	Cite some examples of integrity constraints that you think can apply to da	5	easy	Databases and Database Users
14	Give examples of systems in which it may make sense to use traditional f	20	medium	Databases and Database Users
15	Define the following terms: data model, database schema, database state	20	hard	Database System Concepts
16	Discuss the main categories of data models. What are the basic difference	15	medium	Database System Concepts
17	What is the difference between a database schema and a database state?	5	easy	Database System Concepts
18	Describe the three-schema architecture. Why do we need mappings betv	20	hard	Database System Concepts
19	What is the difference between logical data independence and physical d	15	medium	Database System Concepts
20	What is the difference between procedural and nonprocedural DMLs?	5	easy	Database System Concepts
21	Discuss the different types of user-friendly interfaces and the types of us	20	hard	Database System Concepts
22	With what other computer system software does a DBMS interact?	15	medium	Database System Concepts
23	What is the difference between the two-tier and three-tier client/server	5	easy	Database System Concepts
24	Discuss some types of database utilities and tools and their functions.	15	medium	Database System Concepts
25	What is the additional functionality incorporated in n -tier architecture (r	5	easy	Database System Concepts

Sheet1

Figure 11: Excel Sheet Containing Topics to Be Added to the System

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

CONCLUSION

In conclusion, the question paper generation system is a valuable tool with numerous benefits for educational institutions and instructors. Through its automated capabilities, it streamlines the process of creating customized question papers, saving time and effort. This system ensures a fair and balanced distribution of questions across various topics and difficulty levels, enhancing the overall quality and fairness of assessments.

Moreover, the system provides a user-friendly interface, making it accessible to educators and administrators without advanced technical skills. It also allows for easy customization and adaptation to specific curriculum requirements, ensuring that assessments align closely with educational objectives. However, it is important to consider certain challenges and limitations. The effectiveness of the question paper generation system heavily relies on the quality of the question bank and the algorithms used for paper creation. Ensuring the accuracy and diversity of questions in the database is essential to generate meaningful assessments.

REFERENCES

Aldabe, Lacalle, L. d., Martinez, Matrixalar, & Uria. (2006). ArikIturri: An Automatic Question Generator Based on Corpora and NLP Techniques.

Ele S. I., Akinola O. A., Egete D. O. and Ele, B. I. (2017). Computerization of the Students' Industrial Work Experience Scheme (SIWES) Registration and Payment System (SIWES-RPS) in Nigeria. Scholars Journal of Engineering and Technology (SJET). Vol.5(3). Pp. 104-116.

Ele, S. I. and Adesola, W. A. (2013). Design of Computer Fault Diagnosis and Troubleshooting System (CFDTS). West African Journal of Industrial and Academic.

Ele**, S. I.,** Ofem O. A. and Obono I. O. (2018). Object-Oriented Design (OOD) Consideration in the Design and Deployment of Agent-Based Patient Appointment and Scheduling System (APASS): A Case of University of Calabar Medical Center. In the Saudi Journal of Engineering and Technology (SJEAT). Vol-3, Iss-4 (Apr, 2018): 228-237.

Ele, S. I., Umoh, E. E., Ele, B. I. Obono I. O. (2021). CADBLOS: Android–based UML-centric design approach to campus-drive bus locator. Journal of Mathematics & Computer Science. 11(5), 5897-5915. https://doi.org/10.28919/jmcs/6055. Available online at http://scik.org.

Fenil, K. G., Hital, G. G., & Ashwini, D. (2017). Automatic Question Paper Generator System. InternationalJournal of Computer Applications, 47.

Liu, D., & Wang, J. (2013). Automatic Test Paper Generation Based on Ant Colony Algorithm, 8(10), 2600–2606. https://doi.org/10.4304/jsw.8.10.2600-2606

Mohandas, M., Chavan, A., Manjarekar, R., & Karekar, D. (2015). Automated Question Paper Generator System. International Journal of Advanced Research in Computer and Communication Engineering , 678.

Naik, K., Sule, S., Jadhav, S., & Pandey, J. (2014). Automatic Question Paper Generator System Using Randomization Algorithm. International Journal of Engineering and Technical Research, 194.

Purohit, K. V., Kumar, A., Jabeen, A., Srivastava, S., Goudar, R., Shivanagowda, & Rio. (2012). Design of Adaptive Question Bank Development and Management System. 2nd IEEE International Conference on Parallel, Distributed and Grid Computing.

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB

Saeed, S., Umar, M.S., Ali, M.A., & Ahmad, M. (2014). Fisher-Yates Chaotic Shuffling Based Image Encryption. International Journal of Information Processing, 8(3), 31-41.,

Shajahan, S. (2017). Scope of Automation in Education Sector. Retrieved from Linked-in: https://www.linkedin.com/pulse/scope-automation-education-sector-sagar-shajahan.

Sharmila, D.G., Saranya, S., Rajkumar, N. and Gomathi, K. K. (2016). Novel Architecture for Data – Shuffling Using Enhanced Fisher Yates Shuffle Algorithm. International Journal of Engineering Science and Computing, 6(5), 4932-4935.

Sung, L.-C., Lin, Y.-C., & Chen, C. M. (2006). The Design of Automatic Quiz Generation for Ubiquitous English E-Learning System . Institute of Information Science, Academia Sinica, Taipei, Taiwan.

Tendolkar, S. N., Shirke, R. B., Mayekar, P. S., & Jadhav, S. V. (2017). Automatic Question Paper Generating System for Satisfying Course Objective. International Journal for Research in Applied Science & Engineering Technology, 958.

Wijaya, A. and Chang, F. (2023). Information System Web Design for Class English Proficiency Test Using Fisher Yates Shuffle Method. bit-Tech. 5. 166-173. 10.32877/bt.v5i3.724.

Article DOI: 10.52589/BJCNIT-ZCK1BFKB
DOI URL: https://doi.org/10.52589/BJCNIT-ZCK1BFKB