Real Time Question Paper Generator Using ML

[1]Jayasri B S, [2]Adarsh A, [3]Abhishek A P, [4]Anup Siddu R S, [5]Ramya S

[1]Professor, [2,3,4]BE Student, [5]Assistant Professor
Department of Computer Science & Engineering,
NIE, Mysuru, India

***Abstract*- Contemporary technologies enable teachers to store exam questions in computer databases. Exams are used to measure student learning, which is an essential component of the education system. The development of exam question papers has long drawn attention due to the need to maintain secrecy and adhere to different blooms taxonomy levels. It is a challenge, to address the issue of how these technologies might assist teachers in automatically creating a variety of sets of questions in real time, avoiding repetition, duplication from prior exams while still following the quality of complexity as per blooms level. Hence the idea is to automate the process of generating the exam question papers effectively and efficiently in real time. The system has the ability to instantly generate well-structured and engaging questions with the help of the Naive Bayes algorithm, a randomization technique, and Bloom's taxonomy for varied complexity of marks and levels respectively. Thus, generation of Question paper in real time makes the task of a course instructor considerably simpler and precise when compared to the conventional methods of generating Question paper, which is quite a tedious process and difficult to always maintain secrecy.**

***Index Terms*- Teacher, Exam, Real Time, Question Paper Generation, Difficulty Level, Bloom's Taxonomy, Naive Bayes Algorithm, Random Logic.**

I. INTRODUCTION (HEADING 1)

All Education is essential for success in today's highly competitive society. Tests and exams are essential components of education because they provide reliable measures of students' knowledge and growth. Therefore, itis crucial to create well-designed exam papers and formats. However, creating question papers is a time-consuming and demanding effort for teachers who are worn out by monotonous tasks. By proposing a novel method, this initiative intends to address these issues and lessen the accompanying problems. In our project, test papers are generated using effective technology that randomly chooses questions from a large database. We considerably streamlinethe paper creation process by using this strategy. Our approach overcomes concerns like bias, repetition, and security issues that might result from this subjective process,in contrast to the conventional manual method, where officials design question papers based on their knowledge and style. Despite the fact that the significance of questions is recognized, there are still a few restrictions. Notably,human factors, such as instability and a relatively constrained choice of themes, can degrade the caliber of test papers.

Furthermore, it is difficult to separate teaching from testing because it takes a lot of time and effort for teachersto create exam questions. The use of computers and the automatic creation of exam questions stand out as crucial steps in attaining this desired separation to get around these challenges. Our algorithm chooses questions based on many factors, like department, semester, subject, and marks, using a random mechanism. The machine learning-powered real- time question paper generator system views the production of question papers as its core function. The approach relies on an intelligent and random selection of a varied collection of questions to ensure that the quality of the questions is important in raising the test standards. Importantly, the software creates papers in a way that prevents question repetition, ensuring a novel and original exam experience. However, one may add or temporarily remove questions, chapters, and themes as they see fit. In addition, to maintain the uniformity of the exam, they can define the complexity level of the question paper by choosing the percentage. The question paper will be produced in PDF format for the necessary marks.

II. LITERATURE SURVEY

Swapnil Ghagare et al. [1] developed a system that utilizes a shuffling algorithm to randomize questions sourced from a database. The simplicity and ease of understanding of their algorithm make it easily implementable in other systems. Mrunal Fatangare et al. [2] introduced a question paper generator that offers a solution for selecting from diverse challenging constraints, allowing users to swiftly generate papers. With modules like admin, user, question entry, and question management, the system ensures efficient operation. Noor Hasimah Ibrahim Teo etal. [3] developed a system incorporating text matching and question sorting capabilities. However, a limitation of this system is its restricted capacity to accommodate a limited number of questions. Suraj Kamya et al. [4] presented a system that utilized fuzzy logic to manage various constraints. The fuzzy logic-based technique categorized limitations, enabling easy system modification to accountfor them. This approach enhanced the system's adaptability and flexibility, enabling it to dynamically handle diverse limitations. Vijay Krishnan Purohit et al. [5] proposed an adaptive system based on Bloom's Taxonomy. However, it assumes error-free data entry, which could impact the overall accuracy of the system. Kapil Naik et al [6] implemented a shuffling algorithm in their system to achieve randomization of question papers. The system included various modules for user administration, subject selection, question management, and paper generation. Itsupported multiple languages and allowed the integration of mathematical formulae and diagrams. Although the shuffling algorithm ensured randomness, the system lacked an efficient marking system, resulting in potential question repetition in subsequent papers.

Mr. Amit Sanjay Khairnaret al [7], It mainly deals with the gathering, sorting, and administration of a large number of questions aboutdifferent levels of toughness from scientific as well as non- scientific subjects related to various classes. This paper introduces

the usage of the shuffling algorithm in the Automatic Question Paper Generator System to overcome the mentioned problem. The main part of the shuffling algorithms is to provide randomization techniques in the question paper generation system, so different sets of questions could be generated without repetition and duplication. Guang Cen et al [8], The article presents asystem design for managing automatically generated papers using a B/S architecture and lightweight J2EE utilities. The system consists of modules for user management, subject handling, question entry and management, and papergeneration and management. A robust algorithm is employed to analyze user inputs and create exam papers based on subject, question type, and difficulty level. The system seamlessly integrates with Microsoft Word for question-and-answer revisions and stores completed papers as ".doc" files, offering user-friendly operation, an intuitive interface, high security, stability, and reliability.Nazlia Omar et al [9], The aim of this work is to automate the assessment of exam questions in alignment with Bloom'sTaxonomy. By employing rule-based NLP techniques to identify important verbs and keywords, the studysuccessfully categorizes questions into different cognitive levels. Using a dataset of 100 questions, primarily focused on computer programming, the research demonstrates accurate classification of exam questions according to Bloom's Taxonomy categories. Jayakodi et al [10], Thisstudy addresses the challenge of developing test questions aligned with desired learning goals. It uses Bloom'staxonomy to automatically classify exam questions based ontheir learning levels. Natural language processing techniques like tokenization, stop word removal, lemmatization, and tagging are employed to create a rule set for categorization. Additionally, WordNet and cosine similarity algorithms are used to generate special rules and apply weights. The evaluation process enables evaluators to revise exam papers accordingly.

Mohammed M et al [11], This research utilizes classification algorithms to determine the cognitive levels of exam questions. The experiment consists of two phases. In the first phase, support vector machine (SVM) classifiers are tested using different mappings. The second phase employs Naive Bayes, K- Nearest Neighbours (KNN), and Linear Support Vector Classifier (Linear SVC) with the chosen classifier from the first phase. Results reveal that linear SVC with one-versus- one mapping and tf-idf as the feature extraction technique achieves the highest performance with a 74.8% f-measure. Future work will involve testing the classifier with Malay language exam questions. Syaamantak & Mandal et al [12], This research explores two approaches, LDA and BERT, to determine the cognitive difficulty of assessment questions. LDA, a machine learning method, achieved 83% accuracy in categorizing over 3000 questions. BERT, a deep learning framework, achieved 89% accuracy. The study investigates the influential variables in determining cognitive knowledgeand understanding. Jain M et al [13], It can be difficult to create questions with the right level of balance. Byclassifying educational objectives based on their complexity and specificity, Bloom's taxonomy provides a solution. This study tries to show how Bloom's taxonomy can be used to gauge the difficulty of exam questions. Question papers are categorized into Bloom's taxonomy levels using machine learning techniques. Smith, Matthew & Bull Larry et al [14],proposed a system based on Fuzzy Logic in which all parameters were categorized based upon some logic so that the system can be easily acquainted with them. The drawback of this system was that it could only provide results in analytical and descriptive formats. Munir & Aftab et al [15], Organizations highly value community feedback, and sentiment analysis through social media allows for comprehensive analysis of user-generated text. This paper utilizes Support Vector Machine (SVM) in Weka to perform sentiment analysis, comparing the results using precision, recall, and F-measure metrics on pre- classified Twitter datasets. Results are presented using tables and graphs.

III. Problem Statement

It takes a lot of effort and time to create test questions, and teachers often put in a lot of physical labour throughout this process. The type and calibre of the questions on the testcan also have a significant impact on how well anassessment works. Consequently, the development of "real- time question banks require an automated system that can use machine learning techniques". The methods now used tocreate test questions either rely on established templates or choose randomly from a set question bank. These techniques typically result in question papers that are uncoordinated, unbalanced, and unable to adjust to particular assessment requirements.

1. Understanding the extent of the assessment requires ananalysis of the curriculum and learning objectives.
2. Extracting inquiries from a sizable question bank ordatabase that are diverse and pertinent.
3. Assessing the level of challenge and cognitive abilitiesneeded for each question.

IV. Existing System

In the current manual technique for question paper generation, questions are chosen and assembled for the test by human workers. These professors or teachers select questions depending on the curriculum and syllabi. Thissystem, however, lacks flexibility and could not quite match the goals of the exam. Additionally, writing question papers manually takes time, especially when it involves severaldifferent subjects. In comparison to the suggested automatedsolution, it also raises the possibility of paper leakage. It takes more time and effort to evaluate several questionsbefore choosing which ones to include on the test. The introduction of a Question Paper Generator into the existing system can improve efficiency by automating the paper generation process. Despite covering a significant portion ofthe task, it lacks proper classifications based on department, chapters, and difficulty levels of questions. However, the automated technique allows for quicker question paper generation and allows users to generate test papers rapidly and randomly, saving a significant amount of time. Theproblem of redundant questions is addressed by using various randomization techniques. Furthermore, putting in place such a system lessens the demand for additional manpower.

A. Drawbacks of Existing System

- The system's database is limited in size, which limits its ability to scale for larger test questions and more datasets.
- The system lacks Bloom's taxonomy notions to create the exam questions.
- As the questions are not organized into modules, there is less security because anyone can access thesystem.
- Since the system just uses a shuffling algorithm, the question sheets are not all unique and frequently ask the same questions.
- The system lacks versatility in adopting variousmarking schemes, and it can only generate papers for a narrow range of marks.
- The system is centralized in that it is constrained toa particular infrastructure and cannot be accessedor used from off-site

locations.

V. PROPOSED SYSTEM

The Real Time Question Paper Generator is specialized software made to help schools, institutes, publishers, and those in charge of creating test papers manage a large database of questions effectively and produce test papers with simplicity. It addresses the difficulties involved with gathering, classifying, and planning a large number of questions across multiple subject areas and levels ofdifficulty for numerous groups. The Naive Bayes algorithm is used in this system to classify questions based on complexity level, allowing for the creation of variedquestion sets free of repetition and duplication.

A. Advantages of Proposed System

- The proposed approach will take less time and be more effective than the current system.
- The created question paper will accurately reflect the administrator's input.
- The proposed system is extremely safe because there is no possibility of question paper leakingbecause it is solely dependent on the administrator.
- The resulting test can be modified to meet specifications.
- An easy-to-use user interface that makes updating data go more smoothly.
- In only a few seconds, generate and develop a properly formatted question paper.
- The type of question may be application-based, knowledge-based, memory-based, or logic-based.
- Questions can be easily modified. The automated nature of the suggested approach makes analysis relatively simple.
- The user may immediately and randomly produce test papers, which will save a lot of time.
- Randomizing questions is made possible by algorithms.
- The database may always be expanded with new questions, and there is no restriction on the number of test papers that can be generated.
- Exam papers can be generated using this methodup to a few minutes before the exam, so there is no chance of exam papers being leaked.

VI. TOOLS AND TECHNOLOGIES USED

1. The term "hypertext markup language," or HTML, refers to the language used to construct web pages. While "markup language" refers to the use of tags to specify the page's structure and elements,"hypertext" refers to the use of hyperlinks within an HTML page.
2. The technology known as CSS, or "Cascading Style Sheet," is used to format how web pages are laid out. Text styles, table sizes, and other design elements may now be defined, which was previously only allowed within the HTML code. On the other hand, Bootstrap is a well-liked CSS framework used to create responsive and mobile- first websites. It offers a selection of pre-designed templates and components to speed up web development.
3. Microsoft created C#, often known as "C-Sharp,"an object-oriented programming language for the.NET Framework. It is influenced by C++, Java,Eiffel, Modula-3, and Pascal, among otherlanguages. Developers can design a variety of safe and dependable programmes using the C# programming language, including database apps, online applications, distributed applications, and window applications. C# provides the adaptability and resources required to create diverse applicationtypes to satisfy varied needs.
4. A web framework called ASP.NET makes it possible to create websites and web apps using HTML, CSS, and JavaScript. It has tools for building Web APIs and incorporating real-time systems like Web sockets. As an alternative to ASP.NET, ASP.NET Core offers programmers a flexible and cutting-edge framework for creating reliable web solutions. Developers now have the resources and tools required to produce outstandingwebsites and online apps thanks to ASP.NET.
5. A GUI tool called SQLyog was created for theRDBMS (Relational Database ManagementSystem) MySQL. Users of MySQL now have an easy-to-use interface for interacting with the database engine thanks to this software. In a varietyof contexts, including physical, virtual, and cloud installations, SQLyog's powerful features allow users to administer MySQL and MariaDB servers and databases effectively. SQLyog, despite being primarily created for Microsoft Windows, may also work with Linux and Unix operating systems whenusing the Wine compatibility layer.

A. Development Environment Used

- Microsoft created Visual Studio Code, a flexible source-code editor that is accessible on Windows, Linux, and macOS. It offers a wide range of capabilities, including integrated Git capability, debugging support, syntax highlighting, intelligent code completion, code snippets, and code refactoring. The editor can be altered in a numberof ways by users, including themes, keyboard shortcuts, preferences, and the ability to expand its features by installing extensions that offer extrafunctionality.

- Google Chrome is a popular web browser that can be used on many different platforms and was created by Google. It serves as both a standalone browser and a key part of the Chrome OS, which drives web applications. Google Chrome offers consumers a seamless web surfing experience thanks to its cross-platform interoperability,delivering a number of features and capabilities to boost productivity, security, and customization.

VII. ALGORITHMS USED

Naive Bayes, a classification algorithm, is employed in machine learning. It relies on Bayes theorem and assumes the independence of each feature, hence the term naive. By examiningobserved features, the algorithm computes the probabilities of various class labels. It determines the ultimate classification by combining the probabilities of each feature occurring in each

class.Despite its simplistic assumption, Naive Bayes is renowned for its effectiveness, user- friendliness, and capability to handle extensive feature spaces. This makes it especially well-suited for addressing text categorization issues.

A. *Role of Naive Bayes Algorithm in Question PaperGenerator Model*

The Naive Bayes algorithm can be used to calculate the difficulty levels of questions in the context of automated testquestion generation. The system learns the patterns and traits connected to various difficulty levels by training on a dataset of labelled questions with known difficulty levels. Inorder to draw conclusions about the relationships between these elements and the difficulty ratings, it looks at a varietyof question characteristics, such as language variety andsentence complexity. When creating new questions, the system evaluates their characteristics and determines their level of difficulty using the patterns it has discovered. It is crucial to acquire high-quality and diverse training data in order to ensure the appropriate assignment of difficulty levels. By integrating Naive Bayes into question paper production systems, the process is streamlined, producing fair and applicable test questions for educational environments while also saving teachers' time and effort.

B. *Naive Bayes Algorithm*

Step 1: Scan the dataset (stored in database)

Step 2: Calculate the probability of each attribute value.[n, n_c, m, p]

Step 3: Apply the formulae

$P = [n_c + (m*p)] / (n+m)$

Where:

- n = Total count of particular key word repeated inwhole dataset
- n_c = Total count of keyword combination with eachcomplexity level
- p = Prior estimate for P (Probability value)
- m = the equivalent sample size (Attributes i.e., L1, L2,L3)

Step 4: Multiply the probabilities found (P) by p

Step 5: Compare the values and classify the attribute valuesto one of the predefined sets of class.

C. *Why Naïve Bayes algorithm is better than other algorithm*

Naive Bayes is a strong algorithm that offers scalability, simplicity, and efficiency. It is a preferred option in some areas due to its handling of categorical characteristics, robustness to irrelevant variables, and strong performance with insufficient data. To ascertain whether Naive Bayes is the best method for a certain task, it is essential to thoroughly evaluate the specific characteristics of thedataset and the issue at hand.

D. *NLP*

Natural Language Processing, or NLP for short, is an area of artificial intelligence (AI) that focuses on how computers and human language interact. In order tocomprehend, examine, and produce natural language text or voice, computational models and algorithms must bedeveloped and put to use NLP's main objective is to make it possible for computers to comprehend and interpret human language in a manner that is comparable to that of humans. Understanding grammar, semantics, syntax, pragmatics, anddiscourse are only a few of the linguistic and cognitiveconcepts that are involved. By bridging the gap between human language and machine comprehension, NLP techniques let computers process, interpret, and produce language in meaningful ways.

E. *How NLP is used in automatic question paper generation using ml project*

Tokenization and part-of-speech tagging are essentialnatural language processing (NLP) concepts for text analysis in the context of automatic question paper production utilizing machine learning.

- Tokenization: Tokenization is the process of breaking down a text document into smaller text components, such as words, phrases, or even tokenized sub word bits.
- Part-of-speech tagging: In this technique, each word in a phrase is given a grammatical tag indicating its syntactic category, such as a noun, verb, or adjective. This labelling gives details abouteach word's purpose and place in the sentence.

The automatic question paper production system efficiently processes and analyses text data, identifying meaningful units and understanding the syntactic structure by using tokenization and part-of-speech tagging techniques. This knowledge makes it easier to create queriesthat are grammatically correct and contributes to preserving coherence and clarity. These methods also aid in locating crucial phrases or ideas in the text, improving the precision and applicability of the questions that are generated.

VIII. System Architecture

The initial step of this process involves the creation of a dataset comprising questions for further analysis. These questions undergo pre-processing techniques to ensure data quality by eliminating errors and assessing their complexity levels. To classify the questions, the Naive Bayes algorithm is utilized in conjunction with Natural Language Processing (NLP) concepts. The questions are segmented into individual words based on Bloom's Taxonomy, which facilitates the identification of key words for classification. Subsequently, unnecessary words are removed from the questions.

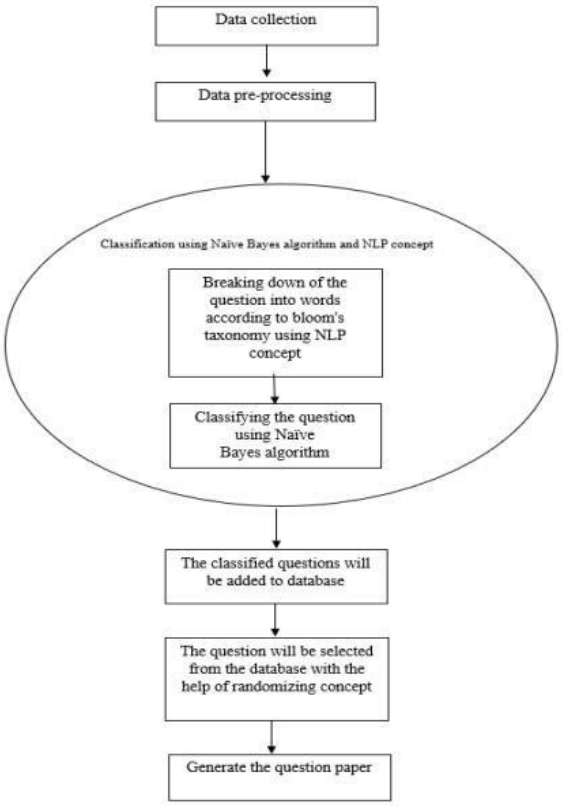

Fig. 1. System architecture

Following the pre-processing phase, the newly obtained questions are classified using the Naive Bayes algorithm,and complexity levels are assigned accordingly. These classified questions are then stored in a database for future utilization. To generate a question paper, the system requiresinput from the user, including the total marks, desired percentages of complexity levels, and the specific chaptersto be included. Once all the necessary details are provided, areal-time question paper is generated using the concept of randomization. Questions are randomly selected from the database, considering the constraints specified by the user. The depicted architecture in the Fig. 1. illustrates this process, showcasing the sequential steps involved in question paper generation, from dataset creation to real-timepaper generation based on user input and randomized question selection from the database.

IX. System Implementation

Our goal during this early stage was to develop a thorough understanding of the project. To do this, we carried out a detailed literature review to find problems with question paper production. Our research found several problems with the current system. With a clear knowledgeof the issues at hand, our attention switched to utilizing automation and machine learning as potential solutions. We also looked at who may benefit from the application and identified its user base. As a result, we were able to develop a creative strategy to achieve our intended objectives. So, after we had a finalized issue statement and a pre-processed dataset, we went on to figure out how to create a real-time question paper generator. This required selecting the right classification algorithms for estimating the level of the question's difficulty and designing a system architecture that took the flow of data into consideration. The project's needs and scalability were taken into consideration when choosing the technology stack to be used. Due to their industry-standard web development capabilities, we used HTML, CSS, and Bootstrap for front-end development. We used SQL Yog and C# for the back-end due to their portability, lightweight design, and superior performance. System Implementation is carried out in two phases: Design Phase and Actual Implementation.

A. Design Phase

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which the interactions take place. These diagrams describe how and in what order the objects in a system function. There are three actors in our system namely: Admin, HOD and Lecturer.

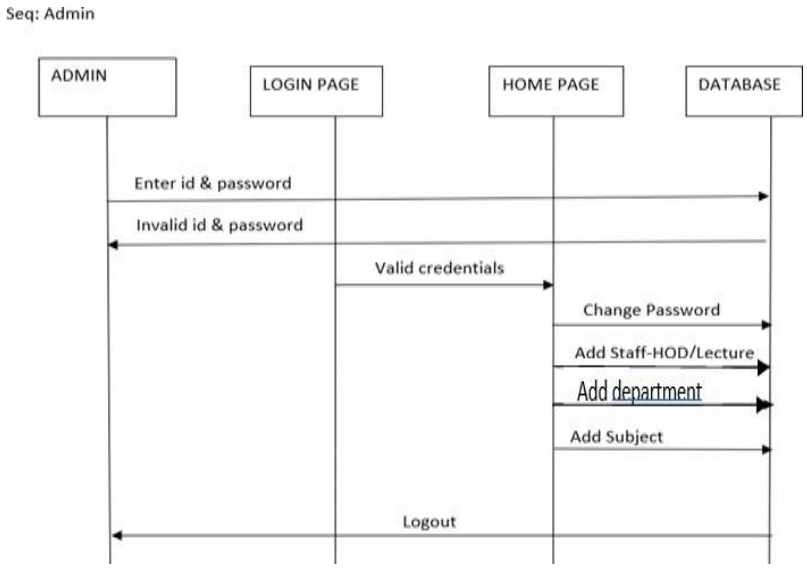

Fig. 2. Sequence diagram for Admin

The detailed sequence diagram for Admin is shown in Fig. 2. The admin will login to the web page with the id and password if both matches with the database, he will be givenaccess to the homepage or he will be redirected back to the same login page. With the valid credential he can change password, add the staff, Hod to the database and he also canalso add department and subjects to the database.

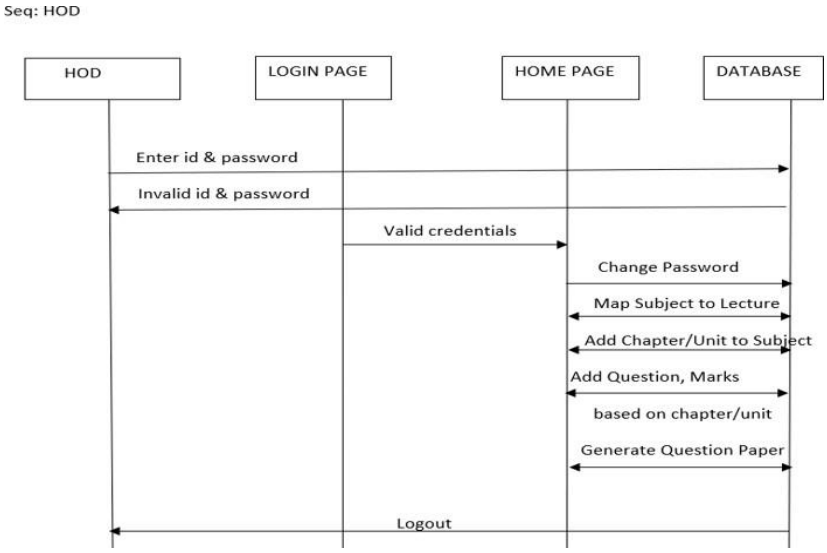

Fig. 3. Sequence diagram for HOD

The detailed sequence diagram for HOD is shown in Fig. 3. The HOD will login to the web page with the id and password if both matches with the database, he will be given access to the homepage or he will be redirected back to the same login page. HOD can also change their password, can map the subject with the other lecture if heis not taking the subject. HOD can add the chapter, questions, marks to the database. Then the questions added in the database will be classified according to the complexity level with the help of ML algorithm. Hod can generate the question paper.

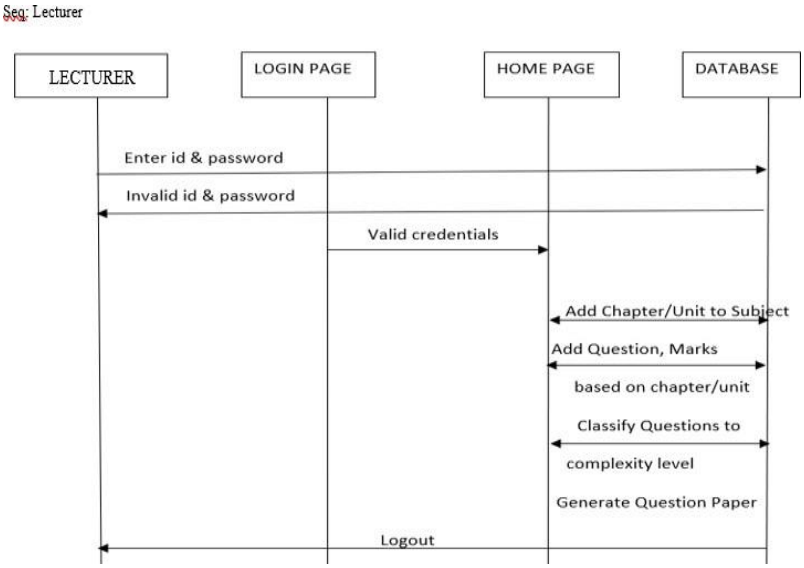

Fig. 4. Sequence diagram for Lecturer

The detailed sequence diagram for Lecturer is as shown in Fig 4. The Lecture will login to the web page with the id and password if both matches with the database, he will be given access to the homepage or he will be redirected back to the same login page. The lecturer can also change the password. The lecturer can add the chapter, questions, marks to the database. Then the questions added in the database will be classified according to the complexity level with the help of ML algorithm. The lecturer can generatethe question paper.

B. Actual Implementation

The three components of the actual implementation are the webpage, the backend database, and the machine learning. Each actor's table, the trained question bank, and the most recent question sets are all included in the database. By using the Naive Bayes algorithm and the taxonomy of blooms, machine learning trains the queries. The questions on the generated test paper are chosen at random based on the predetermined criteria.

The three actors in our system each carry out a unique activity when they check in. The three actors can all alter the randomly generated password, which is a common move.Here is a detailed summary of the steps taken.

Admin:
Add Department.
Add Staff.
Add Subject.
HOD:
Add and delete Chapters.
Map and un-map lecturers.
Add and delete questions.
Generate question paper.
Lecturer:
Add and delete Chapters.
Add and delete questions.
Generate question paper.

All these actions are completed in a hierarchy without jeopardizing security. The criteria for each of these actions are selected from the dropdown box, which separates the data from the database and only retrieves those that are permitted for a specific actor.

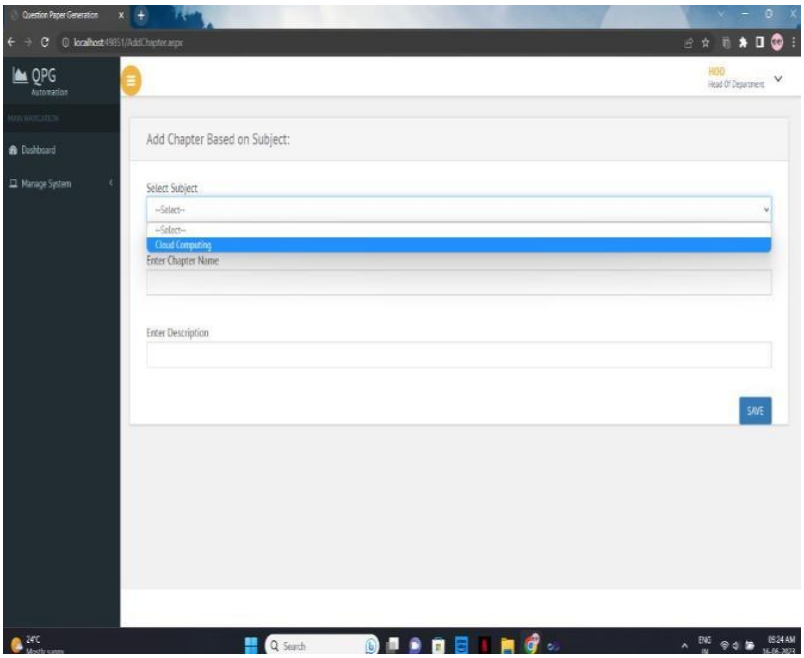

Fig. 5. An actor can only choose a subject if the admin has added it.

In Figure 5, HOD may only add chapters to subjects that admin has previously added, and only if HOD is assigned to that subject.

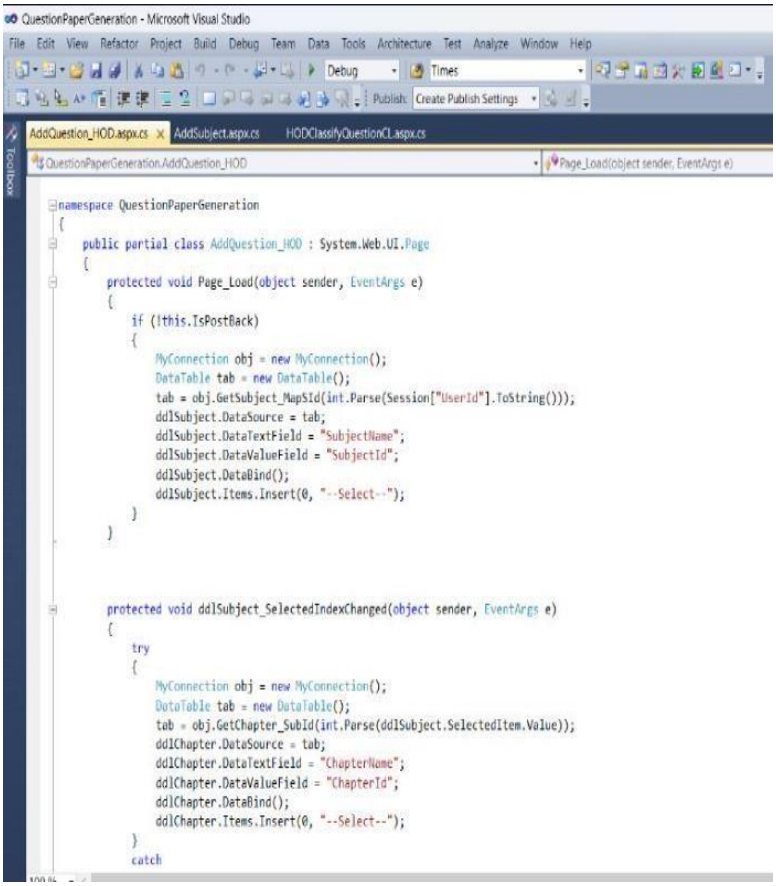

Fig. 6. Snippet of code for HOD question addition.

By choosing the necessary parameters, such as the subject and chapter that have previously been added, HOD can add new questions, as shown in Fig. 6.

X. TESTING

By independently testing programme components, software testing aims to find flaws or faults. Its goal is to check the software's operation and guarantee that the creators invention works as planned and generates the desired results. As a result, testing requires a series of deliberate and organised actions. In this step, the parts are put together to create a full system, and testing is done to show that the system functions well and meets the requirements. Test cases are carefully selected to account for all potential

configurations and evaluate the system's anticipated behaviour. As a result, test cases are chosen that have expected inputs and outputs. Unit testing, system testing, and user testing are among the test methodologies used for software testing.

TABLE 1 UNIT TESTING

Test Case ID	Test Case Name	Test Case Description	Status
TCU01	Test Login Form	Check whether only authorized users can login (i.e., Admin, HOD, Lecturer)	Success
TCU02	Test for Admin to add lecturer	Check whether the admin can add lecturer	Success
TCU03	Test for subjects, modules, questions individually	Check whether the subjects, modules, questions can be added and deleted	Success
TCU04	Test for complexity	Check whether the complexity of questions paper is correct or wrong	Success
TCU05	Test for question paper	Check whether question paper generated is correct	Success

Table 1 lists references to unit testing for each significantuser interface element. It simply tests the component's functioning.

TABLE 2 TESTING COMPLEXITY LEVEL – MARKS COMBINAIONS

Sample Input(marks)	Expected Output(marks)	Obtained Output(marks)	Status
L1 & L2: 25% L3: 75%	L1 & L2: 25% L3: 75%	L1 & L2: 25% (2,3,3,7 - 15/60) L3: 75% (6,6,7,8,6,4,8 - 45/60)	Success
L1 & L2: 50% L3: 50%	L1 & L2: 50% L3: 50%	L1 & L2: 50% (3,2,3,2,12,3 - 25/50) L3: 50% (7,7,6,5 - 25/50)	Success
L1 & L2: 75% L3: 25%	L1 & L2: 75% L3: 25%	L1 & L2: 75% (6,1,3,6,2,2,7,2,8,2,3,3 - 45/60) L3: 25% (7,8 - 15/60)	Success
L1 & L2: 20% L3: 80%	L1 & L2: 20% L3: 80%	L1 & L2: 20% (2,3,3,3,5 - 16/80) L3: 80% (6,6,8,8,9,6,6,7,8 - 64/80)	Success
L1 & L2: 90% L3: 10%	L1 & L2: 90% L3: 10%	L1 & L2: 90% (2,3,2,2,5,2,5,3,2,3,1,8,7-45/50) L3: 10% (5/50)	Success
L1 & L2: 100% L3: 0%	L1 & L2: 100% L3: 0%	L1 & L2: 100% (1,2,6,2,8,8,2,2,6,7,6 - 50/50) L3: 0%	Success

Table 2 provides references for calculating the marks given in the form of percentages for choosing a question based on its level of difficulty. The complexity level assessment for the keywords, which can be L1, L2, or L3, is referenced in Table 3.
Table 2 tests for the combination of L1 and L2 complexitylevel by taking percentage of marks as a parameter.

TABLE 3 TESTING COMPLEXITY LEVELS FOR KEYWORDS

Sample Input	Expected Output	Obtained Output	Status
What	L1	L1	Success
Define	L1	L1	Success
Mention	L1	L1	Success
Explain	L2	L2	Success
How	L2	L2	Success
Discuss	L2	L2	Success
Design	L3	L3	Success
Develop	L3	L3	Success
Illustrate	L3	L3	Success

Complexity level for each keyword is tested in Table 3which can be L1, L2, L3.

XI. RESULTS

To guarantee that the system operates properly in a variety of settings, in a dynamically changing environment, at a variety of times, several tests were conducted. The table displayed in the testing segment clearly shows the outcome of our test cases. The outcome shows that the system operates well and generates the desired results as planned. Trials are used to examine each component separately, and precise findings are discovered. Numerous criteria are required for input, and all authorization is handled without jeopardizing security. The final question paper as in Fig. 7. is produced using a layout that meets the specifications and complexity specified.

THE NATIONAL INSTITUTE OF ENGINEERING,MYSURU-8

(An Autonomous Institute Affiliated to VTU,Belagavi)

1st Internals

Cloud Computing

Times:1 hrs. Total Marks:30

SlNo	ChapNo	CL	Questions	Marks
1.	Chapter 1	L3	Using an illustration, Design different layers of a cloud computing model (e.g., IaaS, PaaS, SaaS) and their respective functions.	6
2.	Chapter 1	L3	Compute the cost of running a cloud-based application that requires 10GB of RAM, 50GB of storage, and 2 vCPUs for 24 hours on a cloud service provider that charges $0.10 per GB of RAM per hour and $0.20 per GB of storage per month.	6
3.	Chapter 1	L1	What is scalability of cloud computing?	1
4.	Chapter 2	L2	How does cloud computing impact data security and privacy? Discuss some measures that can be taken to mitigate these risks.	8
5.	Chapter 2	L1	What are the advantages and disadvantages of using serverless computing compared to traditional compute models?	2
6.	Chapter 2	L1	List the types of virtualization technologies used in cloud computing.	2
7.	Chapter 3	L1	What are the challenges of monitoring a cloud-based infrastructure compared to on-premise solutions?	3
8.	Chapter 3	L1	Mention the security considerations for data in a multi-tenant cloud environment.	2

Fig. 7. Generated Question Paper

XII. CONCLUSION AND FUTURE

This software's main objective is to produce questionnaires using a randomization technique. It runs as a desktop application and generates distinct sets of test questions basedon predetermined restrictions, ensuring reliable results with few mistakes. The approach efficiently incorporates randomness into the process of creating test questions while remaining impartial. Users can easily create excellent question papers with a few simple clicks. The real-timequestion paper generator thus offers a safe and effective option.

For future improvements, it is essential to compile agreater variety of keywords that correlate to various degrees of complexity. The model can be enhanced by increasing theclassifiers rule set and keyword pool. Using a classifier makes it possible to include more rules and keywords whichimproves accuracy. A dedicated database with lots of questions would be helpful for the software to reach its full potential. This large question bank would expand the software's capabilities and improve portability. However, when working with a huge database, security becomes a serious issue. To remedy this, a more secure database systemis implemented, guaranteeing that only people with permission can access the software. The software can keep its integrity while producing excellent outcomes by giving security measures priority.

REFERENCES:

1. Mihir Joisher, Swapnil Ghagare, Mittal Patel, and Ritesh Rathi, "Automatic Question Paper Generation System" International Journal of Advanced Research in computer and communication Engineering (IJARCCE), vol.4 Dec 2015.
2. Mrunal Patangare, Rushikesh Pangare, Shreyas Dorle, Uday Biradar, Kaustubh Kale, "Android Based Exam Paper Generator"

Proceeding of the Second International Conference on Inventive Systems and Control (ICISC 2018).
3. Noor Hasimah Ibrahim Teo, Nordin Abu Bakar and Moamed RezduanAbd Rashid, "Representing Examination Question Knowledge into Genetic Algorithm", IEEE Global Engineering Education Conference (EDUCON), 2014.
4. Suraj Kamya, Madhuri Sachdeva, Navdeep Dhaliwal and Sonit Singh, "Fuzzy Logic Based Intelligent Question Paper Generator" IEEE International Advance Computing Conference (IACC),2014.
5. Vijay Krishnan Purohit, Abhijeet Kumar, Asma Jabeen, Saurabh Srivastava, R H Goudar, Shiwangowda, "Design of Adaptive Question Bank Development and Management System", 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, 2012
6. Kapil Naik, Shreyas Sule, Shruti Jadhav and Surya Pandey,"Automatic Question Paper Generation using Randomization Algorithm," International Journal of Engineering and Technical Research, vol. 2, issue 12, December 2014
7. Mr. Amit Sanjay Khairnar, Mr. Bhagwat Chintaman Jadhav , Mr. Rahul Birhade , Mr. Pramod Patil.Research paper on automatic question paper generation(Volume 4, Issue 9, May- 2017).
8. Guang Cen, Yuxiao Dong,Wanlin Gao,Simon See, Qing Wang.Elsevier article. (Volume 51,Issues 11-12)
9. Nazlia Omar, Syahidah Sufi Haris, Rosilah Hassan, Haslina Arshad, Masura Rahmat, Noor Faridatul Ainun Zainal, and Rozli Zulkifli. "Automated Analysis of Exam Questions According to Bloom's Taxonomy" Procedia - Social and Behavioral Sciences, vol. 59, 2012. doi:10.1016/j.sbspro.2012.09.278.
10. Jayakodi, Kithsiri & Bandara, Madhushi & Perera, Indika & Meedeniya, Dulani. (2016). WordNet and Cosine Similarity based Classifier of Exam Questions using Bloom's Taxonomy. International Journal of Emerging Technologies in Learning (iJET).
11. Mohammed M, Omar N (2020) Question classification based on Bloom's taxonomy cognitive domain using modified TF-IDF and word2vec. PLoS ONE 15(3): e0230442. https://doi.org/10.1371/journal.pone.0230442.
12. Das, Syaamantak & Mandal, Das & Basu, Anupam. (2020). Identification of Cognitive Learning Complexity of Assessment Questions Using Multi-class Text Classification. Contemporary Educational Technology. 12. ep275 10.30935/cedtech/8341.
13. Jain M., Beniwal R., Ghosh A., Grover T., Tyagi U. (2019) Classifying Question Papers with Bloom's Taxonomy Using Machine Learning Techniques. In: Singh M., Gupta P., Tyagi V., Flusser J., Ören T., Kashyap R. (eds) Advances in Computing and Data Sciences. ICACDS 2019. Communications in Computer and Information Science, vol 1046. Springer, Singapore. https://doi.org/10.1007/978-981-13-9942-8_38
14. Smith, Matthew & Bull, Larry. (2003). Feature Construction and Selection Using Genetic Programming and a Genetic Algorithm. 10.1007/3-540-36599-0_21.
15. Ahmad, Munir & Aftab, Shabib & Ali, Iftikhar. (2017). Sentiment Analysis of Tweets using SVM. International Journal of Computer Applications. 177. 975-8887. 10.5120/ijca2017915758.