International Journal of Computer Trends and Technology
ISSN: 2231-2803 / https://doi.org/10.14445/22312803/IJCTT-V67I4P102
Volume 67 Issue 4, 5-15, April 2019


*Original Article*

# Automated Question Paper Generator and Answer Checker Using Information Retrieval Approach

Mansi Palav[1], Pranjal Singh[2], Pratik Vishwakarma[3], Pooja Pandit[4], Prof. Neelam Phadnis[5]

[1,2,3,4,5] *BEBE, Department of Computer Engineering, Mumbai University Mumbai, Maharashtra, India*

***Abstract*** *- This is a challenging span due to the growth in computer science and the demand we face today. Hence, examinations play a vital role in testing students' performance. That is why it is important to have a smart development question model for the growth of students and test their learning skills, thereby keeping a check on student performance. Generating an effective question paper is a task of great importance for any educational institute. The traditional method, where lecturers manually prepare question papers, is very tedious and challenging. Our system allows faculty to generate question papers with random questions, which cover the chapters selected by the faculty. In this study, we propose an automated scoring approach for descriptive answers by usingJaro-Winkler*

***Keywords*** *- Question Paper, Answer Checker, Randomize Algorithm, Jaro-Winkler.*

## I. INTRODUCTION

Generating an effective question paper is a task of great importance for any educational institute. The traditional method, where lecturers manually prepare question papers, is very tedious and challenging. Automated Question Paper Generator and Answer Checker System can reduce time consumption by replacing the conventional method of question paper generation. The system fully automates the process of question paper generation and selective answer checker. The advanced system generates question papers based on a database such that all types of questions and answers such as (MCQs, Theory-based, and objectives.) are stored in a database. The system randomly selects the questions from the database. It generates a question paper such that it covers all the chapters selected and allows the student to attempt the examination and score accordingly. Answer Checker checks the paper and generates marks/scores for the student by finding the difference between a model answer and a student answer by usingJaro-Winkler

## II. LITERATURE REVIEW

This section presents the significant approaches to information retrieval and the techniques used. There is a need for automatic generation of question paper and answer checking for text similarity and understanding. Different models used to evaluate the results of these techniques are also reviewed.

### *A. Background Study*

#### *1. Question Paper Generation*

A literature survey was started to understand the need for automatic generation of the question paper. Many existing LMS support tagging features, but users may not utilize this feature fully. The comparative study shows that Moodle is the best LMS for any educational institution and supports a large number of users. But it allows the user to define only question type. Hence the questions in the repository may have only basic tags or no tags at all. So it becomes overhead for teachers to tag these questions before using them. Properly tagged questions can be efficiently retrieved from the repository. Hence, it is necessary to tag the questions before adding them to the repository. A system that offers generation of the question paper using user-given input parameters considers only a fixed range of values. Our system supports upper and lower bounds for inputs and supports a more granular level of topics than chapters and more question types than only three types offered by this system. We use an automatically tagged question repository as input instead of untagged questions.[5]

#### *2. Answer Checker System*

Different models used to evaluate the results are reviewed.

#### *a. Intelligent Essay Assessor (IEA)*

It uses a statistical model to compare descriptive answers and checks the semantic similarity between two answer sets. It is also used to analyze and score the essay type of answers.

#### *b. E-rater*

E-rater is used to analyze the essay type of answers and specifies syntactical and lexical issues in the text. [3]

***c. C-rater***

C-rater is primarily used for assigning the marks as per the student's correct answers. It also deploys similar kinds of words used in the answer, spelling errors, and syntax variations, checked automatically. [2]

## *3. Text to Text Similarity Approaches*

The primary similarity methods are classified as knowledge-based similarity, corpus-based similarity, and string-based similarity measures. [4]

***a. Knowledge-based similarity***

It applies text-to-text similarity to determine the shortest path of similarity by detecting lexical chains between pairs in a text using the WordNet hierarchy.

***b. Corpus-based Similarity***

It is used to find similarities between words according to the corpus sets. It checks the occurrences of the word in the particular answer.

***c. String-based Similarity***

String-based similarity evaluates the measures of similarity or dissimilarity between two text strings. There are two types of string-based algorithms for evaluating the similarity between the student's answer (SASA) and the model answer (MAMA).

***c.1Character-based similarity***

The character-based similarity is used to determine the distance between two strings and perform minimum operations. Operations include insertion, deletion, substitution, and transposition of a single character.

***c.2Term-based similarity***

The term-based similarity is the distance between two items and somethe of the distances of their corresponding items

***D. Cosine Similarity***

The SASA and MAMA are represented as vectors, where the student's answer and model answer are a set of terms; each term has a weight, which reflects its importance on that MAMA or S.AS.A. There are several ways to calculate this weight, such as the Term Frequency-Inverse Document Frequency (TF-IDF). The (TFTF) refers to the term frequency in the model answer, and the IDF represents the importance of a term concerning the entire corpus. It is calculated by the number of answers in the corpus divided by the number of answers containing a term.

The cosine similarity measure is based on the term weighting scheme, which is the TF-IDF.

It is usually a weighting factor in information retrieval and text mining. The formulas of TFTF, IDF, and TF-IDF are illustrated below as follows:

$$TF = \frac{number\ of\ occurrences\ of\ the\ term\ in\ answer}{number\ of\ terms\ in\ answer}$$

$$IDF = \frac{N}{\log nj+1}$$

Where N is the total number of answers, NJNJ is the number of answers containing the term.

$$TFIDF = TF * IDF$$

The main idea behind this model is to calculate the weight of each term in each answer concerning the entire corpus.

The TF-IDF compares a student's answer vector with a model answer vector using a cosine similarity measure. Cosine similarity measures the cosine of the angle between two vectors. Two vectors of attributes, SASA and MAMA, the cosine similarity and cosine (θ), are represented by using a dot product and magnitude as follows:

$$cosine\ similarity(SA, MA) = \frac{Dot\ Product\ (SA,MA)}{SA*|MA|}$$

Where Dot Product is:

Dot Product (SA, MA) =SA [0] *MA [0] +…..+ SA [n] *MA [n]

And Distances | | and |MA| is defined as:
| | = [0] + [1] + ….. + [ ]
And
|MA| = MA [0] + MA [1] + ….. + MA [n]
After the cosine similarity between the model answers and student answers is calculated, marks are assigned.

## *4. Automated Tagging*

The following four tags were identified for the automatic generation of the question paper based on Bloom's Taxonomy.

| Tags | Values |
|---|---|
| Cognitive Level | Recall, Understand, Apply, Analyse, Evaluate, Create |
| Question Type | Fill in the blanks, Multiple choice, Match the following, True False, Answer in one word, Definition |
| Content | Topics and subtopics from the syllabus |
| Difficulty Level | Low, Medium, High |

***a. Cognitive level Identification***

It is the process of understanding, Percept, Evaluate, Analyse, Recalling, and Creating the functioning of the system

***b. Question-type Identification***

Question-type is to identify the objective or subjective type of questions.

***c. Content Identification***

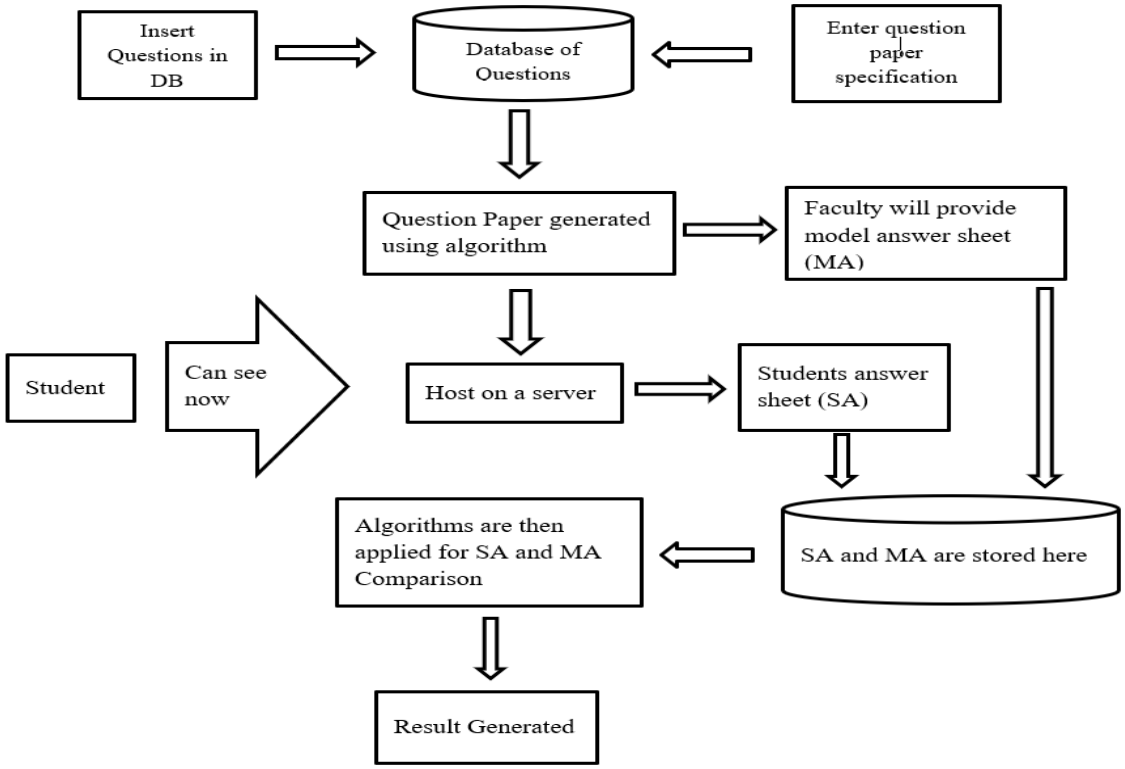
It is to identify the topics as well as subtopics.

***d. Difficulty- Level Identification***

The difficulty level depends on the concept involved, type of question, and cognitive level and decides whether it is low, high, or medium.

## III. PROPOSED SYSTEM

We are presenting an Automated Question Paper Generator System and Answer Checker System that can reduce time consumption by replacing the traditional method of question paper generation.




***System Architecture***

***B. Question paper generation system***

The examiner will input the questions as per their need in their respective database record if they want to manipulate (Add, Delete or Change) the data through a GUI. Questions will get updated as per the function obtained. As this is a web-based application, faculty can set the difficulty level, the structure of the answer, the Chapters they want to add according to their subject, and the Exam Score for how many marks they want to set in their papers. The system will generate random questions about the chapters specified by extracting them from the database using an algorithm. The questions will be well organized.

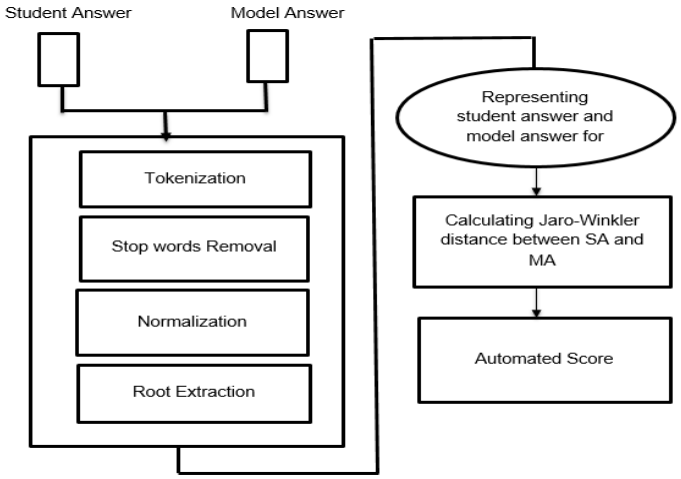
***1. Randomized Algorithm***

A randomized Algorithm checks the duplicate questions and uses them to display random questions. The algorithm is as follows, N = total no. of questions in the database. The randomized algorithm randomly generates these questions.

Step 1: Create an array of N locations.
Step 2: Generate a random number.
Step 3: if (lock==0)
Store generated number.
Else
Compare the generated number with the previous number in the array.
If a matching value is found, go to step 2;
Else
Store the no in the next location.
Step 4: Repeat step 2 for N numbers.
Step 5: Select questions from DBDB matching with values from array location one by one [1].

***B. Answer Checker System***

The student appearing for the test needs to log in and appear for the test (Test will be of Subjective and objective pattern). The more accurate the answers, the more they will earn the marks. The difference between
model answer and student answer are done using Jaro-Winkler, which checks all the possible mistakes that the student has made and gives the final score of the test.




***Answer Checking Process***

***1. Pre-processing***

Pre-processing plays a very important role in answer checking. The pre-processing operations needed are segmentation, stop-word removal, normalization, finding synonyms, and extracting roots.

***a. Tokenization***

Tokenization is needed to identify the end of each sentence. The sentences are ended with various punctuation marks that can be a dot (.), comma (,), colons (:) etc.

***b. Stop-words Removal***

Stop-words are the most frequent words we use in our answers, such as prepositions, articles, and conjunctions and are not much useful for automatic scoring. Removal of these stop-words will improve the performance of the system.

***c. normalization***

Normalization is needed to modify the text to make it definite as per needed by removing unnecessary characters and non-alphanumeric characters to improve the system's performance.

***d. Root Extraction***

The keyword sources are extracted from both student answers and model answers.

***e. Jaro-Winkler distance***

Jaro-Winkler is used to compare strings by measuring the edit distance between two strings. The minimum the distance, the more the strings are similar. The score is normalized such that 0 is an exact match and 1 is for similarity. The jaro-Winkler similarity is 1- Jaro-Winkler distance.

The Jaro Similarity (simj) between two strings s1 and s2 .

$$\text{sim}\,j = \left\{0 \;\; if m = 0 \middle| \frac{1}{3}\left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m}\right) otherwise\right\}$$

Where |Si| = length of the strings

M = number of matching characters

T = is half the number of transpositions.

Two strings, s1 and s2, are considered matching only if it is the same and not greater than

$$\left[\frac{\max(|s1|, |s2|)}{2}\right] - 1$$

Jaro-Similarity uses prefix scale (p), which gives ratings to the matched strings from the very start. So for this Jaro-Winkler similarity will be

$$\text{Simw} = \text{simj} + \text{lp}\,(1 - \text{simj})$$

Where

L is the length of common prefix max up to 4 characters.

P is a constant scaling factor to adjust upwards to have a common prefix. p should not be greater than 0.25. Else the distance will become greater than 1. Standard value of p = 0.1

Jaro-Winkler distance is dw.

$$\text{dw} = 1 - \text{simw}$$

***1.6 Automatic Scoring***

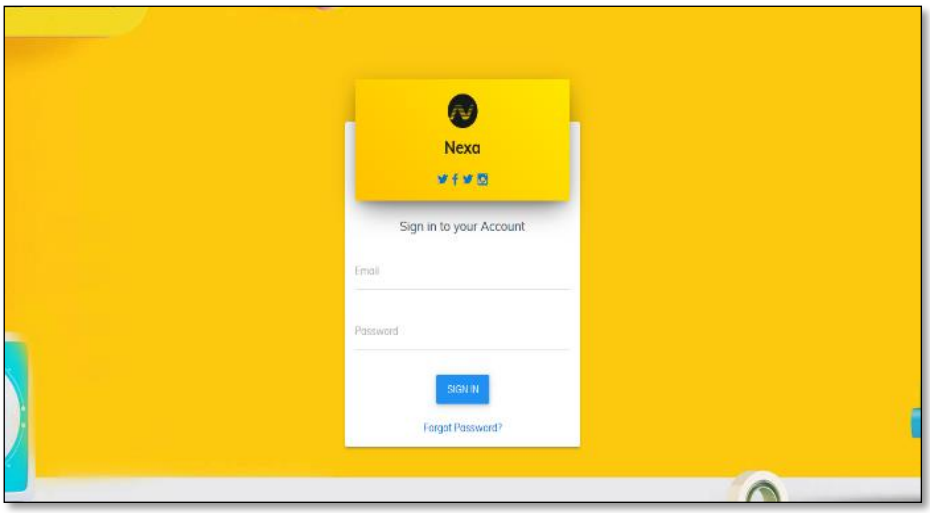
The similarity measure value is converted into a score using the following formula:

Mark = Similarity – Value * Mark

Similarities are calculated by the JaroWinkler method.

## IV. IMPLEMENTATION OF THE SYSTEM

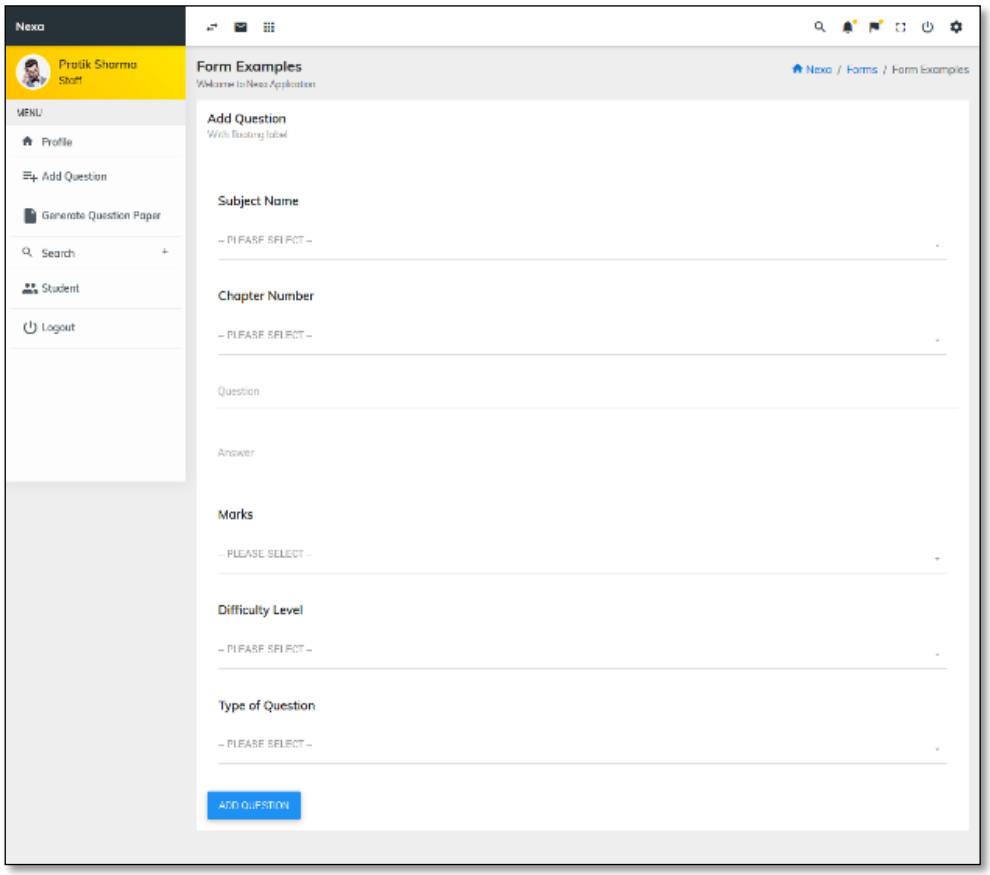
### *1. Login Page*




**Login Page**

The login page is common for admin, staff, and students in our system. By default, all the usernames and passwords are inserted in the database uniquely for all types of users (Admin, Faculty, and Student). The user does not need to register for their session. Only they need to authenticate themselves through login and start with their session, respectively.

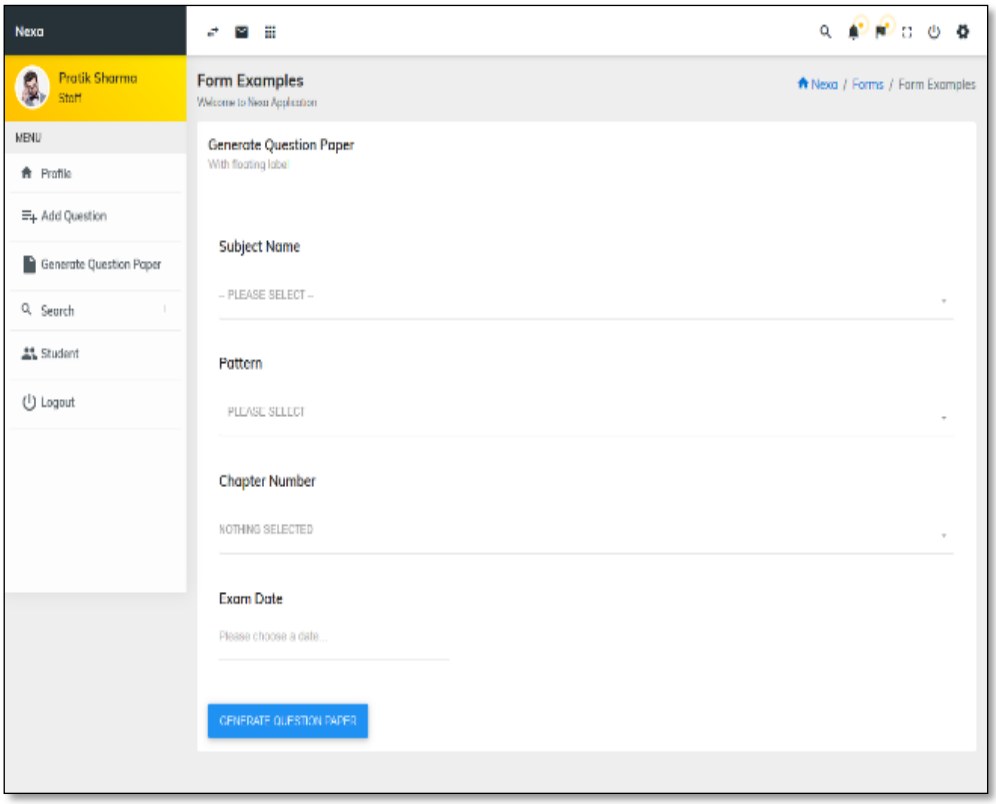
### *2. Add Question and Answer*




The faculty can insert additional questions in the database if required. They only need to enter a subject name, Chapter Number to which the question belongs. Then enter the question and answer as the faculty need to give their model answer in the database. They have to select marks (1, 2, and 5)for that particular

question that they need to allocate. Select the difficulty level (Low, Medium, and High) and the type of question (MCQ, FIB, Define, and Brief).

***3. Generate Question Paper***

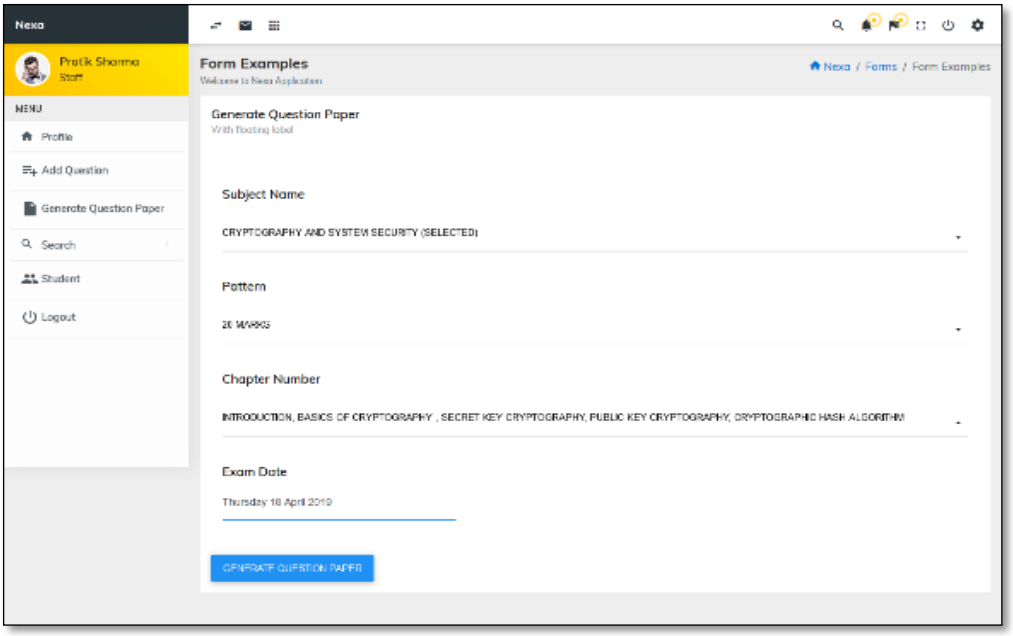



Faculty need to enter the Subject name that they need to generate. The pattern of the paper (20, 40, 60, 80). Chapter names are displayed. Faculty need to select at least two chapters to generate the paper pattern.

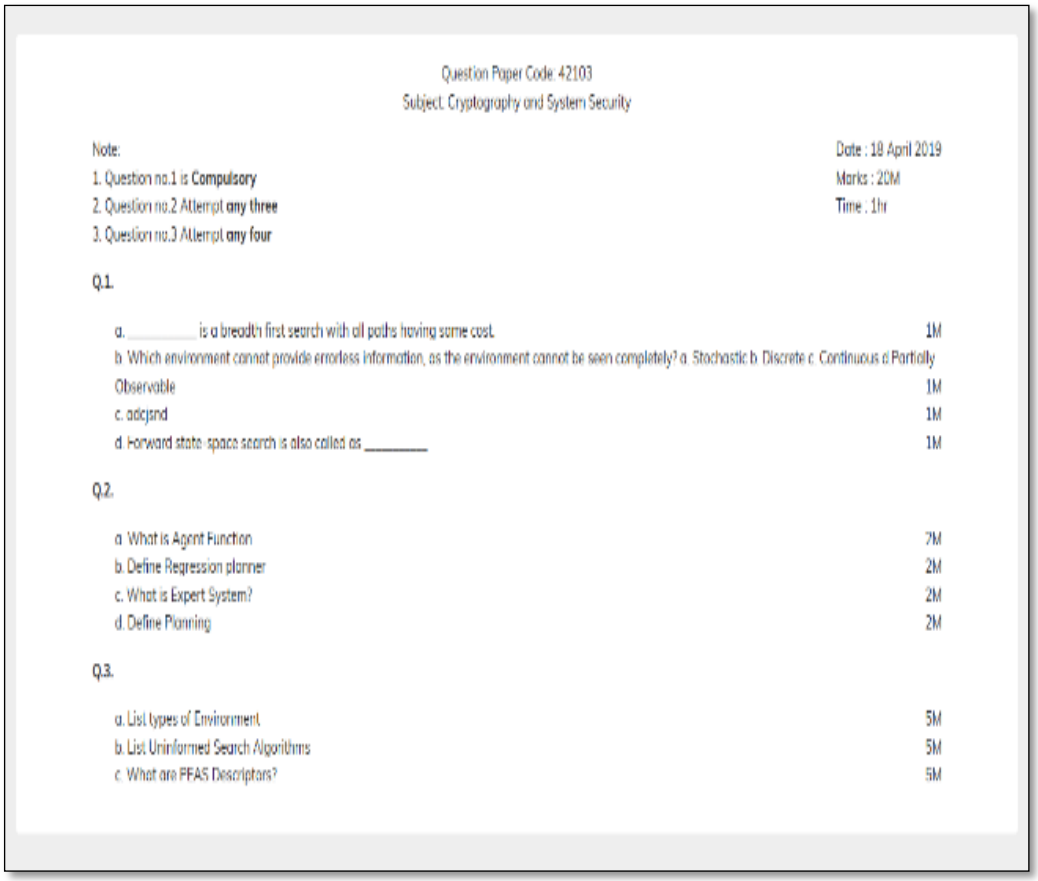
The exam Date is selected when the exam is conducted. By clicking on generate question paper, the question paper is generated with random questions per the selected chapters.

***3.1 Sample Paper Generated***

***Inputs***




***Output***



Question Paper Code: 42103

Subject: Cryptography and System Security

Note:
1. Question no.1 is **Compulsory**
2. Question no.2 Attempt **any three**
3. Question no.3 Attempt **any four**

Date : 18 April 2019
Marks : 20M
Time : 1hr

Q.1.

a. _____ is a breadth first search with all paths having same cost. 1M
b. Which environment cannot provide errorless information, as the environment cannot be seen completely? a. Stochastic b. Discrete c. Continuous d. Partially Observable 1M
c. adcjsnd 1M
d. Forward state-space search is also called as _____ 1M

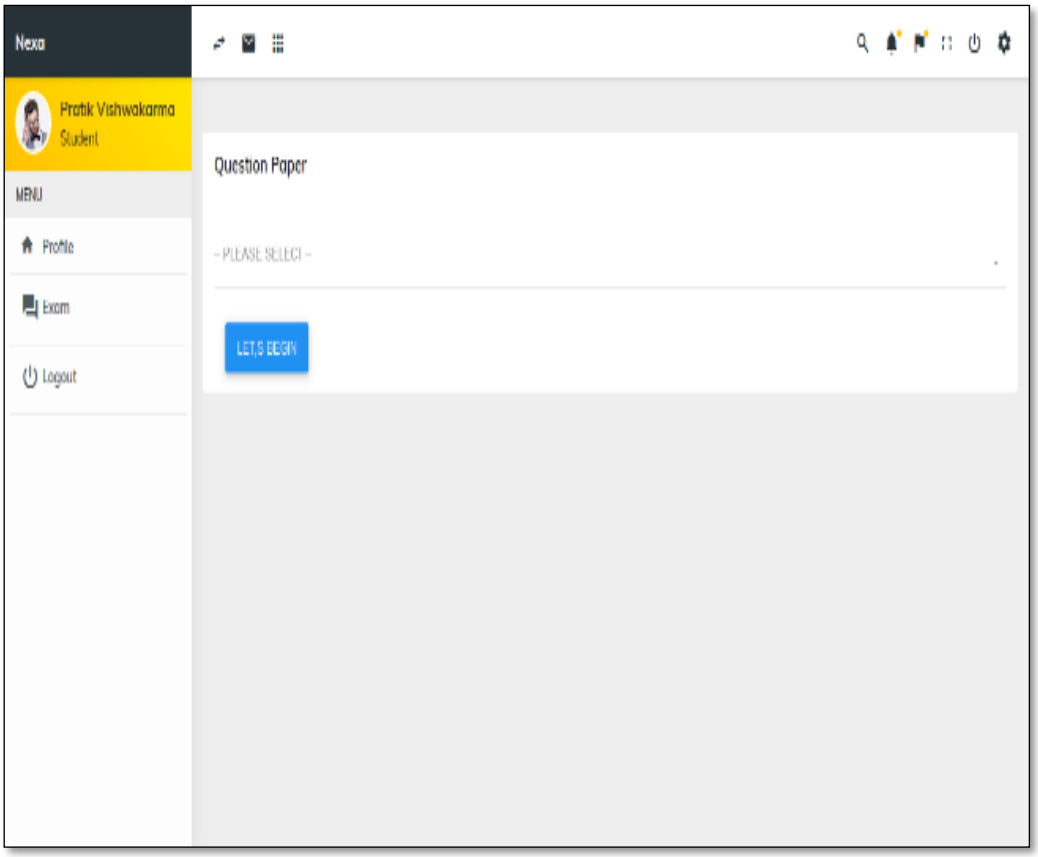
Q.2.

a. What is Agent Function 7M
b. Define Regression planner 2M
c. What is Expert System? 2M
d. Define Planning 2M

Q.3.

a. List types of Environment 5M
b. List Uninformed Search Algorithms 5M
c. What are PEAS Descriptors? 5M

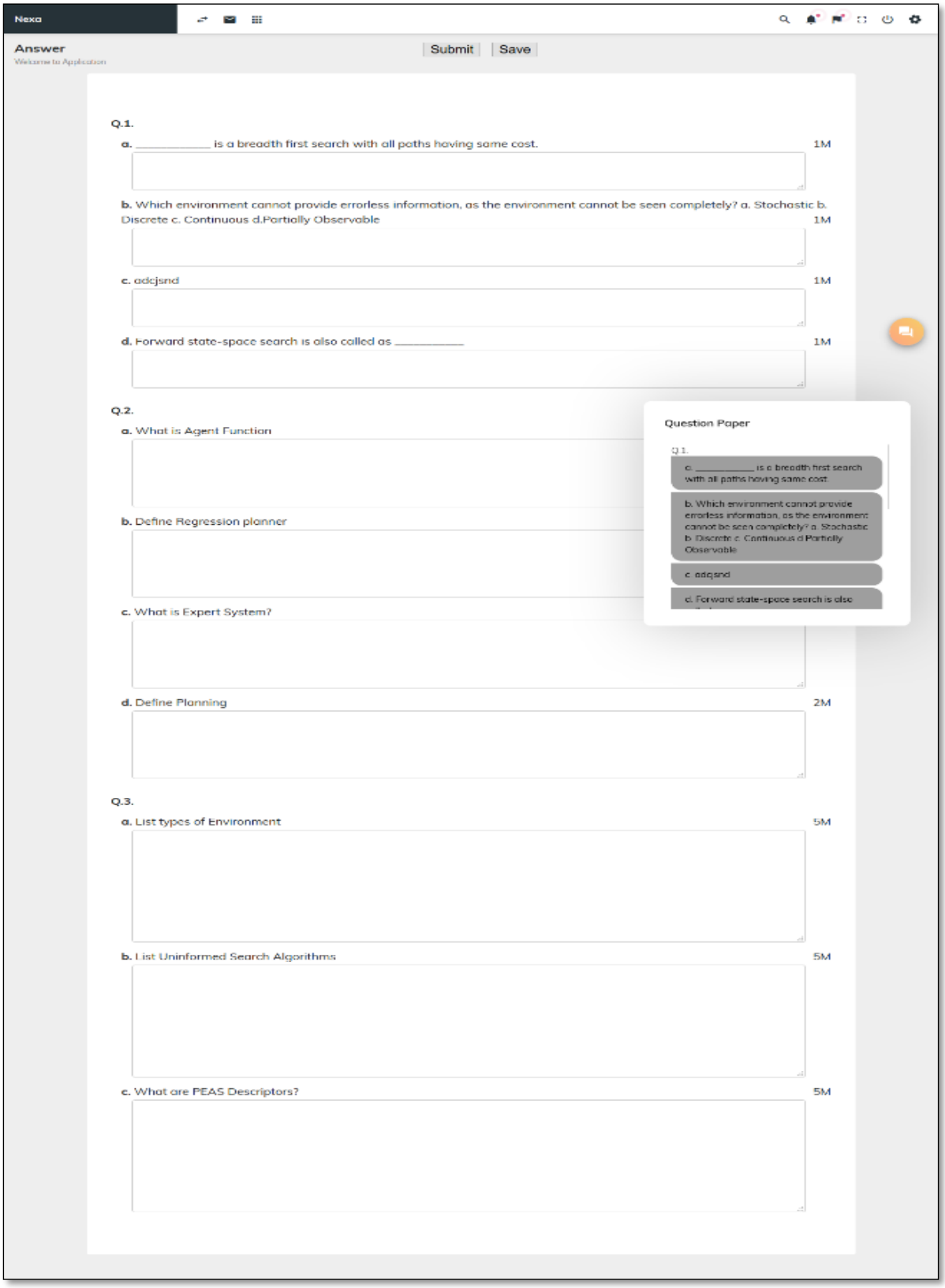
***4. Student- Start Exam***




Students log in and start their session when it needs to appear for the exam. Students need to select the question paper, which is hosted, and then the exam session gets started by clicking on Let's Begin. With

the help of marks of individual questions, total marks are then calculated automatically, and the result is displayed.

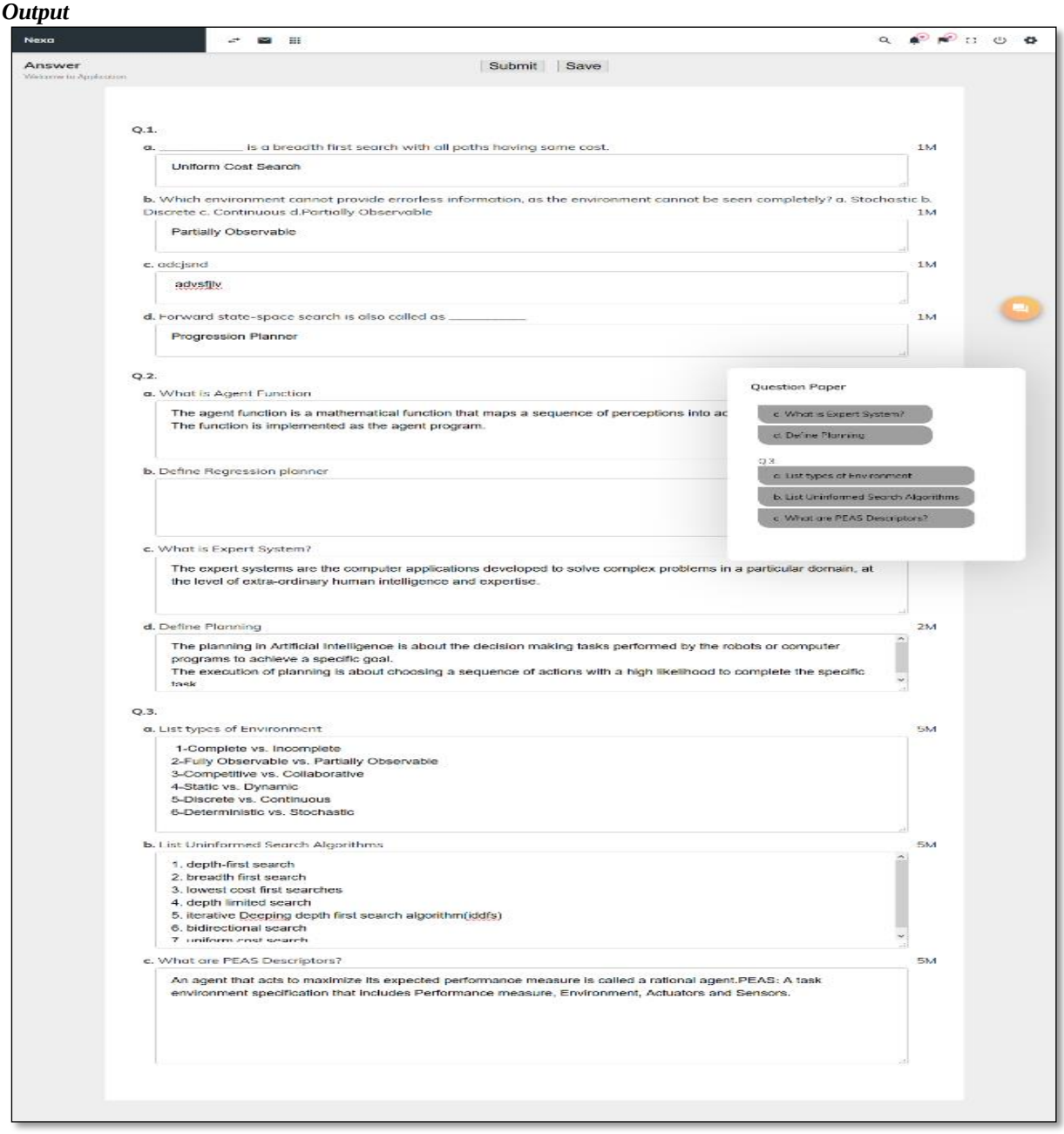
### *5. Answer Sheet*



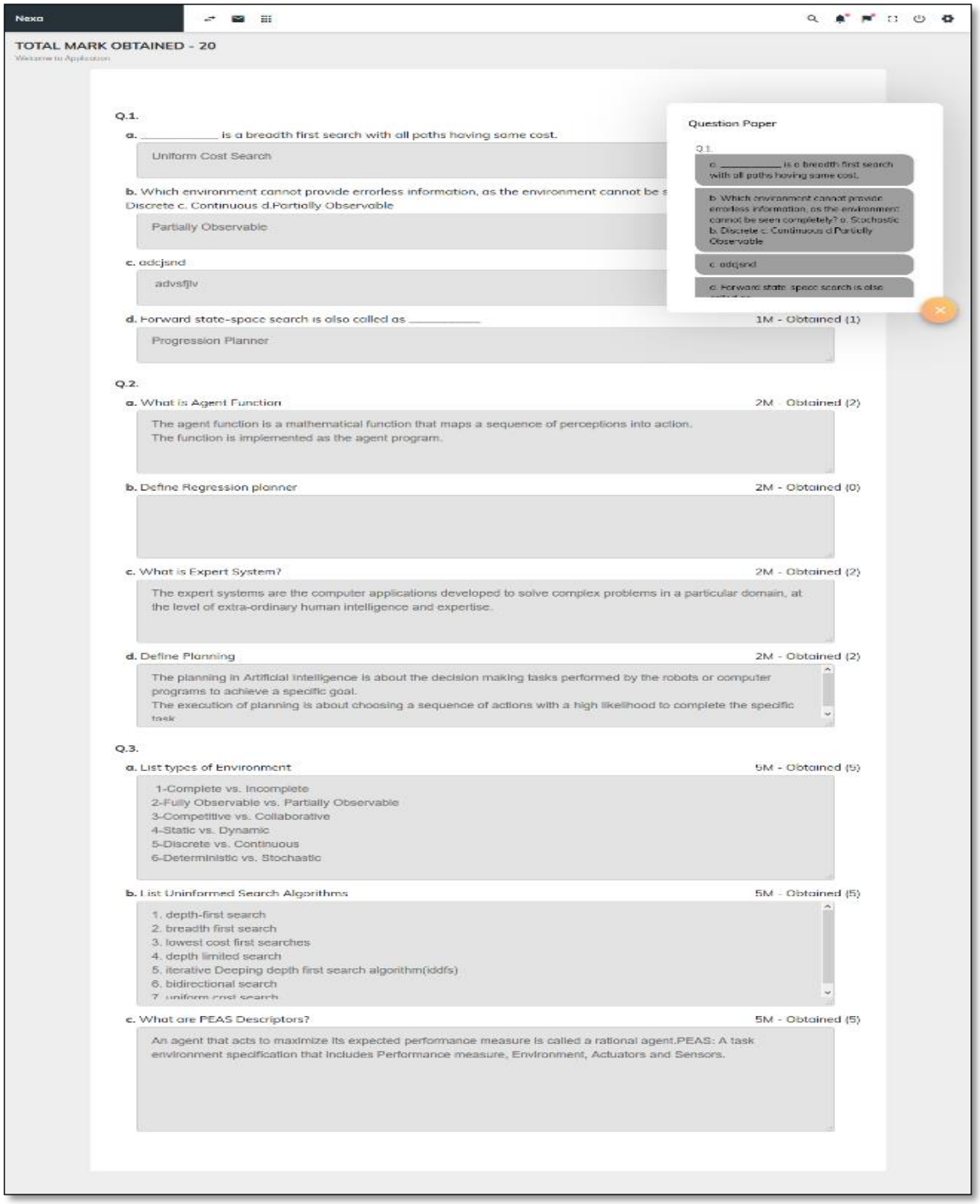
### *5.1 Sample Answer Generated*

***Inputs***


***Output***



***After Submission***




## V CONCLUSION

We have implemented automatic question paper generation using a randomized algorithm. The system can generate 20, 40, 60, and 80 marks of question papers with various types of questions like MCQ's (Multiple Choice Questions), Fill in the Blanks, Answer in one sentence, and Short answers. The answer checking system generates the marks for all the individual questions based on the similarity measure between the student answer and the model answer. This system can be used further in various colleges and schools to reduce their work and effective time utilization.

## REFERENCES

[1] Kapil Naik, Shreyas Sule, Shruti Jadhav, Surya Pandey, Automatic Question Paper Generation System Using Randomization Algorithm, (IJETR) ISSN: 2321-0869, Volume-2, Issue-12, December 2014.

[2] C. Leacock and M. Chodorow, C-Rater: Automated Scoring of Short-Answer Question, Computers and the Humanities, vol. 37, no. 4, pp. 389-405, 2003.

[3] Y.Attali, J. Burstein, Automated Essay Scoring with E- Rater V.2.0, Paper presented at the Conference of the International Association for Educational Assessment (IAEA) Philadelphia, 13 - June.

[4] M.Mohler, R. Mihalcea, Text-to-text semantic similarity for automatic short answer grading, Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, pp. 567-575, March.

[5] Rababah, H., & Al-Taani, A. T. (2017). An automated scoring approach for Arabic short answers to essay questions. 2017 8th International Conference on Information Technology (ICIT).

[6] Noor Hasimah Ibrahim Teo, Nordin Abu Bakar, and Mohamed Rezduan Abd Rashid, Representing Examination Question Knowledge into Genetic Algorithm, IEEE Global Engineering Education Conference (EDUCON), 2014.
.

[7] A Statistical Approach for Automatic Text Summarization by Extraction Munesh Chandra Vikrant Gupta Santosh Kr. Paul 2011 International Conference on Communication Systems and Network Technologies

[8] Deshpande, R., Vaze, K., Rathod, S., & Jarhad, T, (2014). Comparative Study of Document Similarity Algorithms and Clustering Algorithms for Sentiment Analysis, International Journal of Emerging Trends & Technology in Computer Science, Volume 3, Issue 5, September-October 2014.

[9] A New Fuzzy Information Retrieval System Based on User Preference Model Dae-Won Kim and Kwang H. Lee Department of Computer Science Korea Advanced Institute of Science and Technology Kusung-dong, Yusung-gu, 305-701, Daejon, Korea.