

## Automatic Question Paper Generator Using Text Comparison Algorithm

Gopinathan B<sup>1</sup>, Dharunya R<sup>2</sup>, Diana Ranjan D<sup>3</sup>, GokulNandhini T<sup>4</sup>

<sup>1</sup>Associate Professor, <sup>2,3,4</sup>UG Scholars

<sup>1,2,3,4</sup>Department of Computer Science and Engineering

Adhiyamaan College of Engineering (Autonomous), Hosur – 635 130

Tamil Nadu, India

\*\*\*

**Abstract** -Automatic Question Paper Generator is an application that generates question papers automatically. The application allows the concerned authorities (administrators, staffs, and faculties) to upload question banks for different subjects through CSV files, which will be stored in a database. From that database, the application will select the questions randomly for specified subjects according to the number of questions and weightage set, and will generate question papers for the same, as PDFs and Word Documents. The application uses Text Comparison Algorithm, which will assure generation of question papers without duplicities and similarities in the semantics of questions.

**Key Words:**assessment, automatic, generate, prioritization, random, text comparison algorithm, upload

### 1.INTRODUCTION

Assessments are indispensable to evaluate the knowledge of students in every educational institution. In order to conduct assessments, the examiners as well as the staffs will have to set the question papers. This includes syllabus coverage, course outcomes, weightage, and the number of questions. Moreover, care should be taken such that each question paper is set without biases or duplication of questions. This is a tedious and time - consuming task for the staffs and examiners. There may be possibilities of occurrence of errors in manual generation of question papers. This paper describes an automated approach of question paper generation using text comparison algorithm. The main objective is to reduce manual efforts through automation and randomization.

### 2. Body of Paper

The paper on Automatic Question Paper Generator System<sup>[1]</sup> explains generation of question paper by using a randomization algorithm. The application enables the faculties to upload questions with difficulty levels. The question paper is generated based on the difficulty levels, which will be analyzed by the admin, and mailed to different colleges. The paper on Automatic Question Paper Generation<sup>[2]</sup> gives insight on autonomous generation of question papers using Knapsack algorithm. The application allows the admin to add faculty, after which the faculty can upload question banks

through excel files. The faculty can set the difficulty level for the added questions. Knapsack algorithm helps in generating question papers with constraints such as easy, medium, or hard. The question papers are generated as PDFs.

Amit Sanjay Khainar, Bhagwat ChintamanJadhav have proposed an automatic question paper generator application<sup>[3]</sup> using Microsoft Visual Studio and Microsoft SQL. The application generates randomized papers from question banks. Generation is done randomly using SQL queries.

A Survey on Automatic Question Paper Generation System<sup>[4]</sup> has presented an automated approach that features various operations such as selection of subjects, entering of questions, managing paper, generation, and so on. It benefits transportation of question papers to different institutes, yet it suffers from storage issues and modification constraints. Shravari N Tendolkar, Rupali B Shirke have described an automatic question paper application that satisfies the course objectives<sup>[5]</sup>. They have made use of Fuzzy logic and Apriori algorithms for paper generation. The application is implemented in ASP.NET.

From the analyses, we get to know that an automated system for efficient and effective generation of question paper is indispensable, in every educational institution, to minimize manual process, errors, and biases.

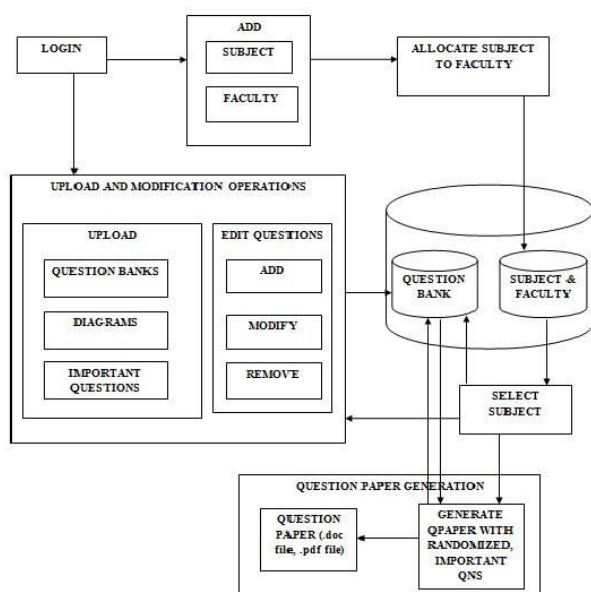
### 3. Proposed System

The proposed system is a web-application developed to generate papers according to the university pattern followed for internal and external assessments. It provides support to input question banks through CSV files, which aids in bulk uploading, thereby saving more time. There are features for modifying, inserting and removing questions as and when needed without any difficulty. The questions can also be prioritized for each subject.

An overview of the proposed system is shown in figure 1.

1. The admin logs into the system, adds a faculty, and allocates a subject to that faculty.
2. The faculty logs in, and he/she can upload question banks for the allocated subject, for each unit.
3. The uploaded question banks are maintained in a database repository.
4. The admin selects the subject and the type of assessment for generating question paper.

5. The application will generate unique and randomized question papers in either word or pdf file.



**Fig -1:** Architecture of Question Paper Generator

## 4. Implementation

Randomized generation of question papers would ensure generation without duplicity. But, a concern is what if the questions generated are similar in semantics, context, or meaning. For example, consider the following:

“What is Automation?” and  
“Define Automation.”

Both the questions have different structures, but the context is the same. In order to avoid such conflicts in generation, we propose Text Comparison Algorithm. This algorithm performs comparison of each question with one another for semantic similarities. The algorithm is given below:

### Text Comparison Algorithm

Step-1: Define an array containing interrogative words,  
qn\_array=array(“What is”, “Define”, “Explain,...”)

Step-2: Specify query for selecting k questions at random,  
qn=array(Randomly selected k Questions)

Step-3: For each Question from i=1 to k:

(i) Convert each question to lowercase,  
l\_case=strtolower(Qn[i])

(ii) Remove all punctuations such as ‘?’, ‘.’, ‘:’, ‘...’,  
rem\_pun=preg\_split(“/[s?.,:]+/”, l\_case)

(iii) Remove all interrogating words using the interrogative word array, qn\_array,  
rem\_qn=str\_replace(qn\_array, rem\_pun)

(iv) Compare the filtered questions with one another,  
if(rem\_qn is similar to rem\_qn+1)  
discard rem\_qn

Step-4: End for

Step-5: Check if the number of questions left in qn[] after steps 3 and 4 is less than k

if(true)  
go to step 2

Step-6: End

## 5. Modules

Automatic Question Paper Generator is implemented in PHP and MySQL. The following illustrates the various modules of the application, along with their operations:

### A. Authenticate Faculty Module

If a faculty wants to access the application, he/she has to be authenticated. The administrator can authenticate a faculty through this module (Figure 2) in two steps:

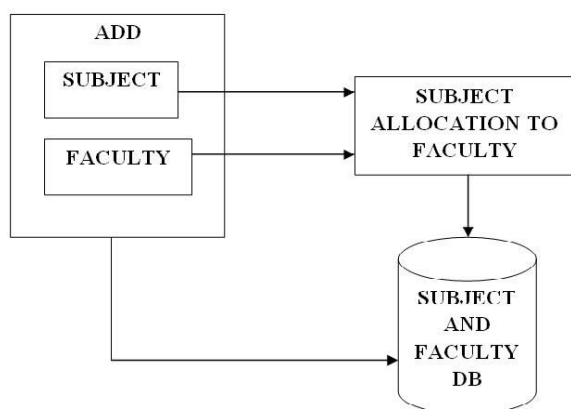
1. The administrator will add a faculty to the database and provide login credentials (username and password).
2. Once the faculty is added, the administrator will select a subject and allocate it to the concerned faculty. Thus, the faculty can then log into the application and upload question banks for the subject allocated.

### B. Add Subject Module

This module enables the administrator to add the subject names along with their course codes for each semester. The application will generate question papers for the selected subject. This is shown in figure 2.

### C. Subject Allocation to Faculty Module

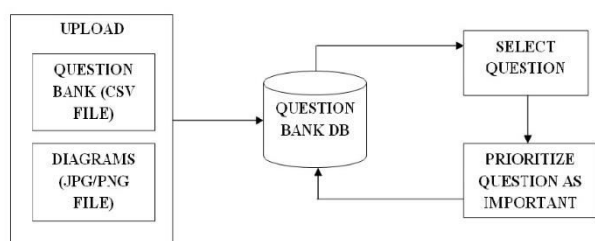
This module allows the administrator to select a subject and allocate it to the concerned faculty, as shown in figure 2. The faculty can upload question banks only for the allocated subject.



**Fig -2:** Faculty Authentication and Subject Allocation

### D. Upload Question Bank Module

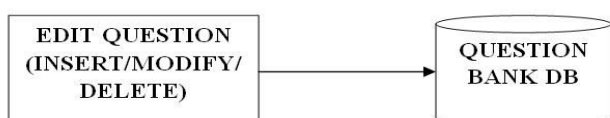
This module allows the faculty to upload question banks for each subject (figure 3). For this purpose, the questions for respective subjects have to be copied and pasted in any editor (Notepad) or excel sheet, and the file is to be saved with .csv extension. The question banks can be uploaded in two sections: one for short questions, and the other for long questions, for each unit. In addition, the faculty can prioritize the important questions as well, for each subject. The module also provides support for uploading diagrams for specified questions, through jpg or png files. For section comprising of long questions, the questions with subdivisions can also be uploaded.



**Fig -3:** Uploading Question Bank

### E. Edit Questions Module

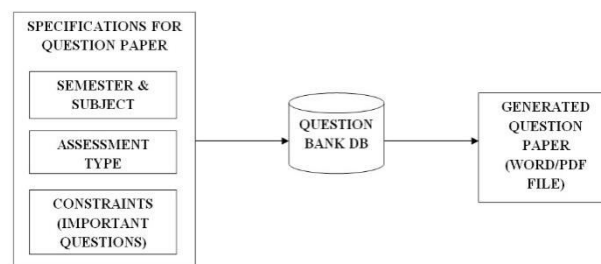
Once the question banks are added, questions can be further added, modified, or removed. This is accomplished through this module, as shown in figure 4. While faculties can add or modify questions, deletion of questions can be carried out only by the administrator.



**Fig -4:** Editing uploaded questions

### F. Generate Question Paper Module

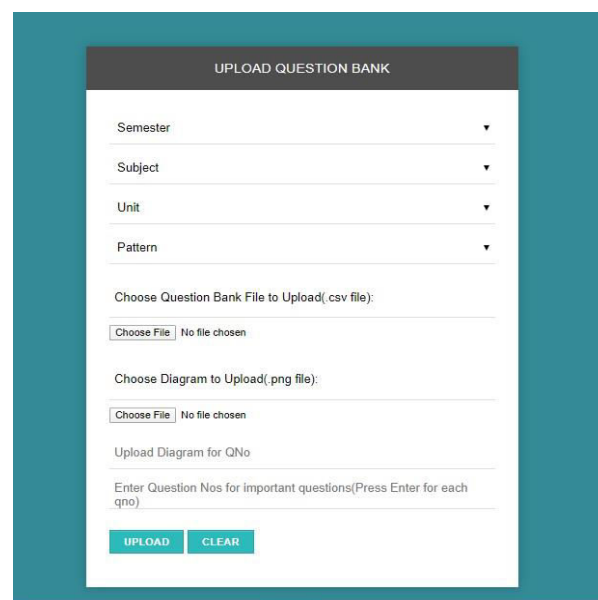
This module depicted in figure 5 is the heart of the application. As the name specifies, the module generates unique and randomized question papers every time without duplicity of questions. This module is accessible by the administrator, who will have to choose the subject and the type of assessment for which the question paper is to be generated. The admin can also specify whether the application has to generate question paper with important questions. The question paper can be generated in both formats .doc, and .pdf.



**Fig -5:** Question Paper Generation

## 6. Results Obtained

The following figure 6 is a screenshot of uploading question bank module. The required fields, semester, subject, unit, and pattern are to be specified, and the question bank file will be uploaded.

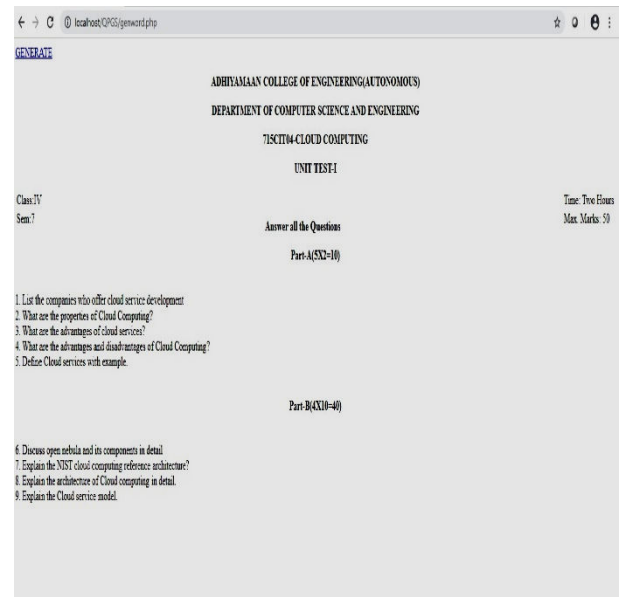


**Fig -6:** Screenshot of Upload Question Bank Module

For instance, if a question bank for the subject Cloud Computing is uploaded, it would get stored in the database as shown in figure 7.

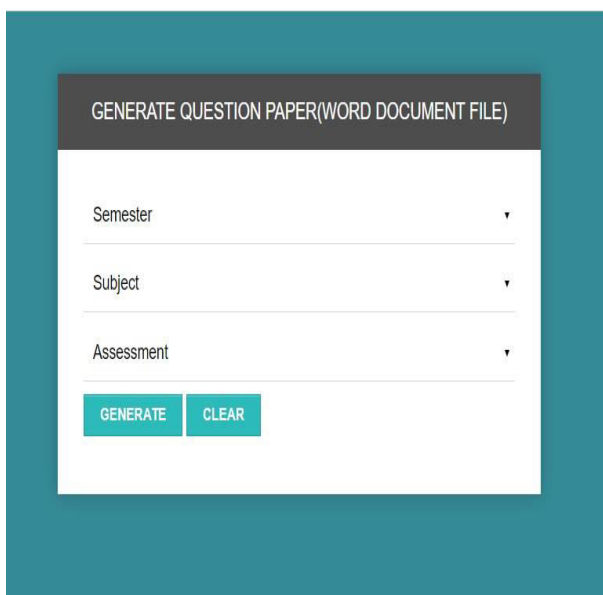


**Fig -7:** Screenshot of Question Bank Database



**Fig -9:** Screenshot of generated questions

Figure 8 is a screenshot of generating question paper module. It requires the administrator to specify the semester, subject, and the assessment.



**Fig -8:** Screenshot of Generate Question Paper Module

Assuming that the question paper for the same subject, Cloud Computing is to be generated for the assessment Unit Test-I, the questions would generate as shown in figure 9.

## 7. CONCLUSIONS

Automatic question paper generating application serves as a helping hand to the examiners and staffs in chalking out question papers. With question banks as inputs, the application will select the specified number of questions randomly, and generate unique, unduplicated, and unbiased question papers with ease and efficiency.

## REFERENCES

1. Zalte. S. V, Jadhav. C. C, Mangire. A. A, Hole. A. D, Tulshi. A. R, "Automatic Question Paper Generator System", IJARCCCE (Vol 7, Issue 3, March 2018), ISSN (e) : 2278 - 1021.
2. Manas Jayde, YashKeer, Prof. SmitaPatil, "Automatic Question Paper Generation", IOSRJEN (Vol 5, PP 25-27), ISSN (e) : 2250-3021.
3. Amit Sanjay Khairnar, Bhagwat ChintamanJadhav, Rahul Birhade, PramodPatil, "Automatic Question Paper Generator", International Journal for Technological Research In Engineering (Vol 4, Issue 9, May 2017), ISSN (e) : 2347 – 4718.
4. AmrutaUmardand, Ashwini Gaikwad, "A Survey on Automatic Question Paper Generation System", International Advanced Research Journal in Science, Engineering and Technology (Vol 4, Special Issue 4, Jan 2017), ISSN(e) : 2393 - 8021.
5. Shrivari N Tendolkar, Rupali B Shirke, Priyanka S Mayekar, Santhosh V Jadhav, "Automatic Question Paper Generating System for Satisfying Course Objectives", IJARSET (Vol 5, Issue X, Oct 2017), ISSN : 2321 – 9653.
6. VinayakHegde, Lavanya V Rao, Shivali B S, "The Framework for Web-Based Automated Online Question Paper Generator through JEE", International Journal of Engineering and Technology.
7. SurbhiChoudhary, Abdul Rais Abdul Waheed, ShrutikaGawandi, Kavita Joshi, "Question Paper Generator System", IJCST (Vol 3, Issue 5, Sep - Oct 2015).
8. P V Bobade, S SDandge, M A Pund, "A Study of Different Algorithm for Automatic Generation of Question Paper", International Science and Technology Journal, ISSN : 1632 – 2882.