

Question Bank Maker

Dr. Arun Thomas¹, Sheethal Shaji², Shery Shaju², Silji Simon C², Vishnu Narayan V²

¹Faculty, ²UG Student

^{1,2}Department of Computer Science & Engineering,

^{1,2}Sahrdaya College of Engineering Technology, Thrissur, Kerala, India

How to cite this paper: Dr. Arun Thomas | Sheethal Shaji | Shery Shaju | Silji Simon C | Vishnu Narayan V "Question Bank Maker" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-3, April 2019, pp.1592-1595, URL: <https://www.ijtsrd.com/papers/ijtsrd23208.pdf>



IJTSRD23208

Copyright © 2019 by author(s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



ABSTRACT

The project is a question bank maker .It is helpful for the students to prepare themselves for learning important topics for the exam . Referring the previous question papers is a part of exam preparation. Learning the same after covering all the topics may be late in most of the cases. Thus if we could sort out and arrange all the questions in module wise it will be easier to go through previous questions. the question bank maker , sorts the Questions module wise, Year wise , Exam wise and mark wise. Question bank maker will produce the questions sort wise easily for the welfare of the students . The students will be able to go through previous questions easily as a part of exam preparations. In future the question bank maker can be fully automated to get question papers after exams identify chapters or module of each questions by itself with machine learning and generate files automatically after exams. Question bank maker is more than a simple question maker or question creator. Here you can upload the question papers as a pdf format and we can sort the question papers in module-wise, mark- wise and exam-wise.

KEYWORDS: Question bank maker, automation, OCR sorting, Google cloud projects

I. Introduction

This "Question bank maker" project is quiet different from the already existing techniques. when we hear this name first we will think that this project is for making questions for the exam. But its not like that .In the light of referring the literature reviews we understand our project should be more different from all that. So in this project we are sorting the questions that more convenient to the users .This will more helpful for students.

Referring the previous question papers of an exam is an important step of exam preparation. Learning the same topic again and again after covering all the topic may be difficult in most of the cases. Thus if we could sort out and arrange all the questions in module wise it will be easier to go through previous questions. Here comes the Question bank maker, Sorts the Questions module wise ,Year wise , Exam wise and marks wise

This project is used to generate question bank from question papers to the students who are suffering to study the portion having high marks at the time of study holidays. The question bank will contain module-wise sorted questions. The question paper is to be submitted and it is processed to generate questions in each subject in module-wise. It contain all the subjects mainly concentrating engineering students. The different branches like Computer- Science, Civil, Bio Medical, Electrical, Electronics Engineering are mainly considered and benefited.

II. Motivation of the project

The question bank maker is an environment in which the faculties can help the students in previous year question papers. The students are able to focus on important portions which should be covered during the last time of study. Both the extraordinary and average students are benefited with the question bank maker. In the initial stage the students can upload the question paper in a pdf format and they are solved by the faculties in module-wise and given back as pdf format.

The main problem today the students facing is unaware of important topics. Even though they have covered all the portions the weightage must be given to the module having higher mark weightage. The difficulties in exams can be solved by this system.The further updation with artificial intelligence will be included in the upcoming versions of this project. This project could be even extended to identify modules by itself with the help of machine learning. Thus students just need to upload question papers to generate question banks. They will also get the old questions with the uploaded ones in generated PDF.

III. Methodology

Through articles and literature reviews we have developed some of the existing methodologies and designed our system based on the difficulties faced by the existing users. Manually creating question papers by sorted manner is the method

adopted here. The existing system is related to manually made question papers, there the question papers are handwritten and the questions are written module wise.

That made people very difficult to develop handwritten question papers. In order to avoid these problems our question bank maker provide a space to users to reduce their efforts. Here the users can upload the question papers as pdf format and the ocr process will be carried out for character recognition, in this version module-wise sorting is done by the faculty and they are uploaded back as pdf. In the upcoming version question can be sorted module-wise by machine learning.

A. Existing Methodologies

1. Manual method

- Manual methods are used by faculties specific to certain subjects they are handling
- Automation can be implemented for only needy systems which may reduce cost

Disadvantages of this existing systems are given below

- Time consuming
- High chance of errors
- Contains minimum number of automated systems
- Tedious Task
- However in our system these disadvantages are overcome effectively.

B. Proposed System

The proposed system integrates all individual systems under one board. So that the overall system will be efficient. The block diagram of proposed system is shown in Fig 2.

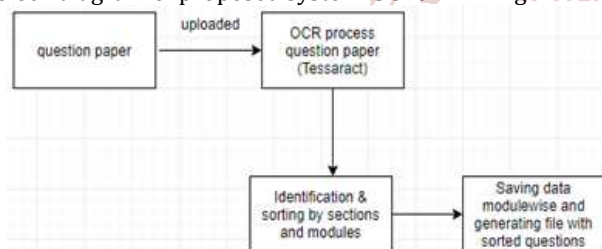


Fig.2. Block diagram of proposed system

The proposed system contains the following subsystems

- Automatic Sorting
- PDF Generation
- Database of Old Question Papers

Automatic sorting

In the question bank maker the automatic sorting can be implemented using machine learning and artificial intelligence. The questions can be sorted in module-wise by automatic sorting. Artificial is used to solve complex problems. The artificial intelligence effect is used here. The main thing is decision making. The machine learning is by studying different algorithms and models that we are using to sort our question papers in module-wise. The patterns and inferences are mainly considered here. Machine learning is a sub branch of artificial intelligence which is mainly used in this system.

PDF generation

The pdf files are generated after automatic sorting. The pdf that is given by the user as an input in an unsorted way is send back as sorted pdf in module-wise. Thus the users are

benefited with the questions that are arranged in module-wise. The pdf are uploaded by the faculties after sorting. This is easily accessible to the students in an efficient way. Using the user interaction and server interaction the pdf's are generated efficiently. The pdf template used here is normal pdf format. The pdf are stored with name of questions papers fir better access of the students.

Database of old question papers

The database of old question papers are stored for future uses. The repetition of question paper that are uploaded by the students and repetition of same questions can be recognized using the database. This will reduce the effort of further sorting the same questions. The same questions need not be sorted further module-wise. It will make the work easier. The backup of questions for further use will be simpler using the database. The mango dB is the database we are using here. As it is a cross platform document oriented database program. By database it is easy ti relate repeated questions

C. Procedure Flow

There are 6 modules in Question bank & the Various modules are

- Upload Module
- OCR interface
- Question identification Form Data input
- Sorting
- PDF File generation

• Upload Module

In the upload module, user can upload the previous question papers in PDF format.

• OCR interface

OCR (optical Character Recognition) system converts the image text content to direct text content converting the printed characters into digital text, allowing you to search for or edit your document in a word processing program. With the OCR image file could be converted to digital text

• Question identification

Analysing and identification of each questions individually from question paper.

• Form Data input

The analysed questions will be represented a form where the user can input chapter or module number

• Sorting

According to the entered chapter or module number the questions are sorted

• PDF File generation

The sorted questions are then reproduced into a PDF file. Thus Question bank is generated. The generated PDF will contain matched subject question from previous exam question papers also.

IV. Programming

The programming is done in Node Js software. During programming we have to keep in mind the code is to be integrated with google cloud.

```
// RUN PACKAGES
```

```

var http = require('http');
const express = require('express'); var fileapp = express();
var fs = require('fs');
const multer = require('multer');
const bodyParser = require('body-parser');

// SETUP APP
fileapp.use(express.static( dirname + '/public')); var server1
= fileapp.listen(3000);

fs.readFile('./QB/index.html', function (err, html) {

if (err) throw err; http.createServer(function(request,
response) {
response.writeHead(200, {"Content-Type": "text/html"});
response.write(html); response.end();
});
});

const app = express();
const port = process.env.PORT || 5000;
app.use(bodyParser.urlencoded({extended:false}));
app.use(bodyParser.json());
app.use('/', express.static( dirname + '/public'));

//MULTER CONFIG: to get file photos to temp server storage
const multerConfig = {

//specify diskStorage (another option is memory) storage:
multer.diskStorage({

//specify destination
destination: function(req, file, next){
next(null, './public/photo-storage');
},

//specify the filename to be unique filename: function(req,
file, next){ console.log(file);
//get the file mimetype ie 'image/jpeg' split and prefer the
second value ie 'jpeg'
const ext = file.mimetype.split('/')[1];
//set the file fieldname to a unique name containing the
original name, current datetime and the extension.
next(null, file.fieldname + '-' + Date.now() + '.'+ext);
}
}),

// filter out and prevent non-image files. fileFilter:
function(req, file, next){
if(!file){
next();
}

// only permit image mimetypes
const image = file.mimetype.startsWith('image/'); if(image){
console.log('photo uploaded'); next(null, true);
}else{
console.log("file not supported")
//TODO: A better message response to user on failure.
return next();
}
}
};

/* ROUTES
***** /

app.get('./QB/index.html', function(req, res){
res.render('./QB/index.html');
});

app.post('/upload',
multer(multerConfig).single('photo'),function(req, res){

res.send('Complete! Check out your public/photo- storage
folder. Please note that files not encoded with an image
mimetype are rejected. <a href="index.html">try
again</a>');
});

// RUN SERVER
app.listen(port,function(){
console.log(' Server listening on port ${port}');
});

void loop() {
• Parsing Code
// RUN PACKAGES
const express = require('express');

var fs = require('fs');
const multer = require('multer'); var S = require('string');
var http = require('http');
const vision = require('@google-cloud/vision').v1; const
bodyParser = require('body-parser');
const {Storage} = require('@google-cloud/storage');

var MongoClient = require('mongodb').MongoClient; var
urldb = "mongodb://localhost:27017/parse";

// Creates a client //google ocr
const client = new vision.ImageAnnotatorClient();

// Bucket where the file resides
const bucketName = 'ktuquestionbank';

// SETUP APP
const app = express();
const port = process.env.PORT || 3000;
app.use(bodyParser.urlencoded({extended:false}));
app.use(bodyParser.json());
app.use('/', express.static( dirname + '/public')); const
projectId = 'firebase-tech2plus';
const storage = new Storage({ projectId: projectId,
});

//MULTER CONFIG: to get file photos to temp server storage
const multerConfig = {

//specify diskStorage (another option is memory) storage:
multer.diskStorage({

//specify destination
destination: function(req, file, next){ next(null,
'./public/photo-storage');
},

//specify the filename to be unique filename: function(req,
file, next){ console.log(file);
const ext = file.mimetype.split('/')[1];

```

```
//set the file fieldname to a unique name containing the  
original name, current datetime and the extension.  
next(null, file.fieldname + '-' + Date.now() + '.'+ext);  
  
}  
}),
```

```
// filter out and prevent non-image files. fileFilter:  
function(req, file, next){  
if(!file){  
next();  
}  
}
```

```
// only permit image mimetypes const image =  
file.mimetype.startsWith('application/pdf'); if(image){  
console.log('photo uploaded'); next(null, true);  
}else{  
console.log("file not supported")  
//TODO: A better message response to user on failure.  
return next();  
}  
};
```

```
app.post('/upload',  
multer(multerConfig).single('photo'),function(req, res)
```

```
const filename = 'C:/Users/vichu  
roxx/Desktop/QBM/public/photo-storage/' +  
res.req.file.filename;
```

```
const onlinefileurl =  
'http://ktuquestionbank.storage.googleapis.com/' +  
res.req.file.filename;
```

V. Acknowledgment

This is an opportunity to express my sincere gratitude to all. At the very outset, we express our thanks to the Almighty God for all the blessings endowed on us. This report is submitted in regard with the project done as a part of the fifth semester curriculum.

We express our wholehearted gratitude to our project guide who give inspiration and suggestions throughout the project work. We extend our sincere gratitude to our project coordinator for leading the way for the completion of the Project. We would like to extend our appreciation to all other faculty members for their help and advices.

REFERENCES

- [1]. International Journal of Engineering and Technical Research (IJETR)ISSN: 2321-0869 Automatic Question Paper Generation System using Randomization Algorithm
- [2]. International Journal of Computer Science Trends and Technology (IJCST) – Volume 3 Issue 5, Sep-Oct 2015 Question paper generator system
- [3]. International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321- 8169 Volume: 4 Issue: 1 173 – 175 Automated Question Paper Generator
- [4]. International journal of emerging technologies in Learning Questions-Bank System to Enhance E-Learning

