

# **AUTOMATIC QUESTION PAPER GENERATOR**

Submitted in partial fulfillment of the requirements for the award of  
Bachelor of Engineering Degree in Computer Science and Engineering

By

**SARAVANAN.J (Reg. No. 39290091)**

**SIVAPRIYAN.M (Reg. No. 39290096)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF COMPUTING**

**SATHYABAMA  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,  
CHENNAI – 600119, TAMILNADU**

**MARCH 2022**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)



---

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafade work of **SARAVNAN.J** (Reg. No. 39290091) and **SIVAPRIYAN.M** (Reg. No. 39290096) who carried out the project entitled “**AUTOMATIC QUESTION PAPER GENERATOR**” under my supervision from December 2021 to March 2022.

Internal Guide

Dr. MOHANPRASAD

---

Submitted for Viva voce Examination held on \_\_\_\_\_

Internal Examiner

External Examiner

## **DECLARATION**

I, **SARAVANAN.J (Reg. No. 39290091) AND SIVAPRIYAN.M (Reg. No. 39290096)** hereby declare that the Project Report entitled "**AUTOMATIC QUESTION PAPER GENERTATOR**" done by me under the guidance of **Dr. MOHANPRASAD** is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science.

**DATE:**

**PLACE: CHENNAI**

**SIGNATURE OF THE CANDIDATE**

## **ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D.**, Dean, School of Computing and **Dr. S. VIGNESHWARI, M.E., Ph.D.**, and **Dr. L. LAKSHMANAN, M.E., Ph.D.**, Head of the Department, Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. MOHANPRASAD**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways For the Completion of the project.

## **ABSTRACT**

The project title "**AUTOMATIC QUESTION PAPER GENERATOR**" deals with the various levels of project development and will account for time used in analysis, design, programming, testing and verification. It is well known fact that generating a question paper manually is a very difficult process and it needs more time and man power. So, here we are proposing an Intelligent Automatic Question Paper Generator System which provides storage of the data, fast operations, and high security for all its tasks

The interface of the project work entitled "**AUTOMATIC QUESTION PAPER GENERATOR**" is developed using **HTML, CSS, JAVASCRIPT, PHP** as Front end and **SQL** as the back end.

## TABLE OF CONTENTS

S.NO	CONTENTS	PAGE NO
1.	<b>INTRODUCTION</b>	
	1.1 OVERVIEW OF THE PROJECT	1
	1.2 MODULES USED IN THIS PROJECT	1
2.	<b>SYSTEM SPECIFICATION</b>	
	2.1 HARDWARE REQUIREMENTS	2
	2.2 SOFTWARE REQUIREMENTS	2
	2.3 SOFTWARE DESCRIPTION	2
3.	<b>SYSTEM STUDY</b>	
	3.1 EXISTING SYSTEM	7
	3.2 PROPOSED SYSTEM	7
	3.3 CHARACTERISTICS OF PROPOSED SYSTEM	7
4.	<b>SYSTEM DESIGN</b>	
	4.1 MODULE DESCRIPTION	9
	4.2 DATABASE DESIGN	9
	4.3 DATA FLOW DIAGRAM	10
	4.4 ENTITTY RELATIONSHIP DIAGRAM	12
	4.5 TABLE STRUCTURE	15
	4.6 INPUT OUTPUT DESIGN	18
5.	<b>SYSTEM TESTING AND SECURITY</b>	
	5.1 SOFTWARE TESTING	20
	5.2 SECURITY	23
6.	<b>SCOPE FOR FUTUTRE ENHANCEMENT</b>	24
7.	<b>CONCLUSION</b>	25
8.	<b>BIBILIOGRAPHY</b>	26
	<b>APPENDIX</b>	
	A. SCREENSHOTS	27
	B. SOURCE CODE	30

## **INTRODUCTION**

### **1.1 OBJECTIVES**

Traditionally question papers were generated manually. Preparing any exam paper is a very challenging task for the educators because they have to check whether there is any repetition in the pattern or not, and other than that security is one of the major concerns for them, also due to lack of teaching staff in any institute, creation of paper is not at all an easy task.

So, here we are proposing an Intelligent Automatic Question Paper Generator System which provides storage of the data, fast operations, and high security for all its tasks. It can be helpful to many educational as well as NGO institutes.

Tasks performed by this system are automated and because of that storage space, security is not a concern anymore. Proposed system works upon Natural Language Processing and is fast due to computer Based automation, streamlined, randomized and unbiased, secure and generates unique questions and overcomes the problems of Human based paper generation.

### **1.2 MODULES USED IN THIS PROJECT**

In this project we use only two modules as follow.

- The first module is importing questions.
- The second module is the pattern seating and mark designing. The third module is the question paper viewing and printing.

## **2.SYSTEM SPECIFICATION**

### **2.1 HARDWARE REQUIREMENTS**

The Software is developed in the system having following configuration.

Processor: 11<sup>th</sup> Gen Intel® Core™ i5-1135G7 @ 2.40GHz

2.42GHz Ram: 8G

Solid State Drive : 512GB

### **2.2 SOFTWARE REQUIREMENTS**

Front End: HTML, CSS, JavaScript,

PHP Back End: My SQL.

Operating System: Windows

10. Software: XAMPP

Server.

### **2.3 SOFTWARE DESCRIPTION**

#### **XAMPP SERVER**

**XAMPP** is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, Maria DB database, and interpreters for scripts written in the PHP and Perl programming languages.

#### **Features of Front End**

The part of a website that the user interacts with directly is termed the front end. It is also referred to as the 'client side' of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and navigation menu. HTML, CSS, and JavaScript are the languages used for Front End development. The structure, design, behavior, and content of everything seen on browser screens when websites, web applications, or



mobile apps are opened up, is implemented by front End developers.

Responsiveness and performance are two main objectives of the Front

End. The developer must ensure that the site is responsive i.e. it appears correctly on devices of all sizes no part of the website should behave abnormally irrespective of the size of the screen. Some of frontend development tools are HTML, CSS and Java Script

## **Features of Front End**

The part of a website that the user interacts with directly is termed the front end. It is also referred to as the 'client side' of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and navigation menu. HTML, CSS, and JavaScript are the languages used for Front End development. The structure, design, behavior, and content of everything seen on browser screens when websites, web applications, or mobile apps are opened up, is implemented by front End developers. Responsiveness and performance are two main objectives of the Front

End. The developer must ensure that the site is responsive i.e. it appears correctly on devices of all sizes no part of the website should behave abnormally irrespective of the size of the screen. Some of frontend development tools are HTML, CSS and Java Script.

### **as a general-purpose programming language. PHP is now installed on more than 244 million Features of Back End**

Backend is the server-side of the website. It stores and arranges data, and also makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a frontend application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend. Some of backend development tools are PHP, Java, C++, Python and Mysql.. etc. Overview of Front End

### **Overview of Front End**

#### **PHP (Hypertext Pre-Processor)**

PHP is a server-side scripting language designed for web development but also used websites and 2.1 million web servers. Originally created by Rasmus Lerdorf in 1995, the reference implementation of PHP is now produced by the PHP Group. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Pre-processor, a recursive acronym. PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page. PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command line interface capability and can be used in standalone graphical applications. PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) due to restriction on the usage of the term PHP. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

#### **Features:**

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS, etc...)
- PHP is compatible with almost all servers used today (Apache, IIS etc...)

- PHP supports a wide range of databases.
- PHP is free. Download it from the official PHP resources: [www.php.net](http://www.php.net).
- PHP is easy to learn and runs efficiently on the server side.

## **HTML5**

HTML5 is a mark-up language used for structure and presenting contents for the World Wide Web and a core technology of the Internet. It is the fifth revision of the HTML standard (created in 1990 and standardized as HTML 4 as of 1997). Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices (web browser, parsers etc...) HTML5 is intended to subsume not only HTML 4, but also XHTML1 and DOM level 2 HTML, HTML5 is also cross-platform. It is designed to work whether you are using a PC, or a Tablet, a Smartphone, or a TV.

### **Features:**

- New features should be based on HTML, CSS, DOM and JavaScript
- The need for external plug-in (Like Flash) needs to be reduced.
- Error handling should be easier than in previous versions.
- Scripting has to be replaced by more mark-ups.
- Some of the most interesting new features in HTML5 are:
- The <canvas> elements for drawing.
- The <video> and <audio> elements for media playback
- Support for local storage
- New form controls, like calendar, date, time. Email, URL, search.

## **CSS CASCADE STYLE SHEET**

Cascading Style Sheet (CSS) is a style sheet language used for describing the look and formatting of a document written in a mark-up language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG, and CSS is a cornerstone specification of the web and almost all web pages use CSS style sheet to describe their presentation.

CSS helps web developers to create a uniform look across several pages of a website. Instead of defining the style of each table and each block of text within a page's HTML, commonly used style needs to be defined only once in a CSS document. Once the style is defined in cascading style sheet, it can be used by any page that references the CSS file. CSS makes it easy to change styles across several pages at once.

### **JavaScript**

JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications.

#### **Features**

- Giving the user more control over the browser
- Detecting the user's browser, OS, screen size etc...
- Performing simple computations on the client side
- Validating the user's input
- Handling dates and time
- Generating HTML code on-the fly without accessing the Web server.

## **Overview of Back End**

### **MYSQL**

MySQL is an open-source relational database management system. It is based on the Structured Query Language (SQL). Which is used for adding, removing and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT and UPDATE can be used with MySQL. MySQL can be used for a variety of applications, but is most commonly found on the web servers. A website that uses MySQL may include web pages that access information from a database.

These pages are often referred to as “dynamic” meaning the content of each page is generated from a database as the page loads. Websites that use dynamic web pages are often referred to as database driven web sites.

Many databases driven websites that use MySQL also use a web scripting language like PHP to access information from the database. MySQL commands can be incorporated into the PHP code, allowing part or all of a web page to be generated from database information. Because both MySQL and PHP are both open source (meaning they are free to download and use), the PHP/MySQL combination has become a popular choice for database driven websites.

#### **USES:**

MySQL is popular for web application and acts as the database component of the LAMP, BAMP, MAMP, and WAMP platforms (Linux/BSD/Mac/Windows- apache-MySQL- PHP-/Perl/Python), and for open-source bug tracking tools like Bugzilla. Its popularity for use with web applications is closely tied to the popularity of PHP and Ruby on Rails, which are often combined with MySQL, PHP and MySQL are essential components for running popular contents management systems such as Drupal, e107, Word Press and some Bit Torrent trackers.

## **3.SYSTEM STUDY**

### **3.1 EXISTING SYSTEM**

Existing system is a difficult process to prepare the Question paper and analysis the data in the textbook.

#### **DRAWBACKS**

- It is a human process.
- In this process repetition in paper may occur.
- Less Security.
- It is a slow process.
- It has less varieties of questions.

### **3.2 PROPOSED SYSTEM**

This software can be widely used in educational institutes etc. It is a web Based application so user location doesn't matter. Admin can access this software anywhere anytime. This system also evaluates the candidate's capability and skills efficiently. It is a fully automated system which results in fast results. In this system there is no need of transporting paper through police/security vans to all colleges. This system provides unbiased results. The usage of this system reduces human effort and saves time and resources to an extent. This project can be further extended to generate yes/no type questions and fill in the blanks type questions. Further automatic answer copy checking systems can be added for checking the answer sheets and providing the result to the student.

### **3.3 CHARACTERISTICS OF THE PROPOSED SYSTEM**

- The Automatic Question Paper Generator System is developed using the Natural Language Processing, Html, CSS, Php and My SQL. In a fully functional system, there is a repository of syllabus, questions and pattern of question papers. It takes a simple text, a document or a pdf file as an input and provides a list of questions as an output



- It is an automated process.
- It is a random as well as unbiased process.
- Higher security.
- It is a faster process

- It is an automated process.
- It is a random as well as unbiased process.
- Higher security.
- It is a faster process.
- It has more varieties of questions.

## **4.SYSTEM DESIGN**

### **4.1 MODULE DESCRIPTION**

#### **IMPORT QUESTION MODULE:**

- \* In this module, we will enter the questions of the particular subject to the database.
- \* We will select the subject and enter the questions to each section.
- \* The entered questions will be stored in the database.

#### **PATTERN SETTING AND MARK DESIGNING:**

- \* In this module, we will select the subject, enter the subject code, and enter the number of questions and number of questions to be answered in sections A, B, C.
- \* Then we will set marks for each section, and select the total marks.
- \* And then click the generate button.

#### **QUESTION PAPER VIEWING AND PRINTING:**

- \* This module is the output page, this page displays the question paper
- \* We can view and print the question paper in this module.

### **4.2 DATABASE DESIGN**

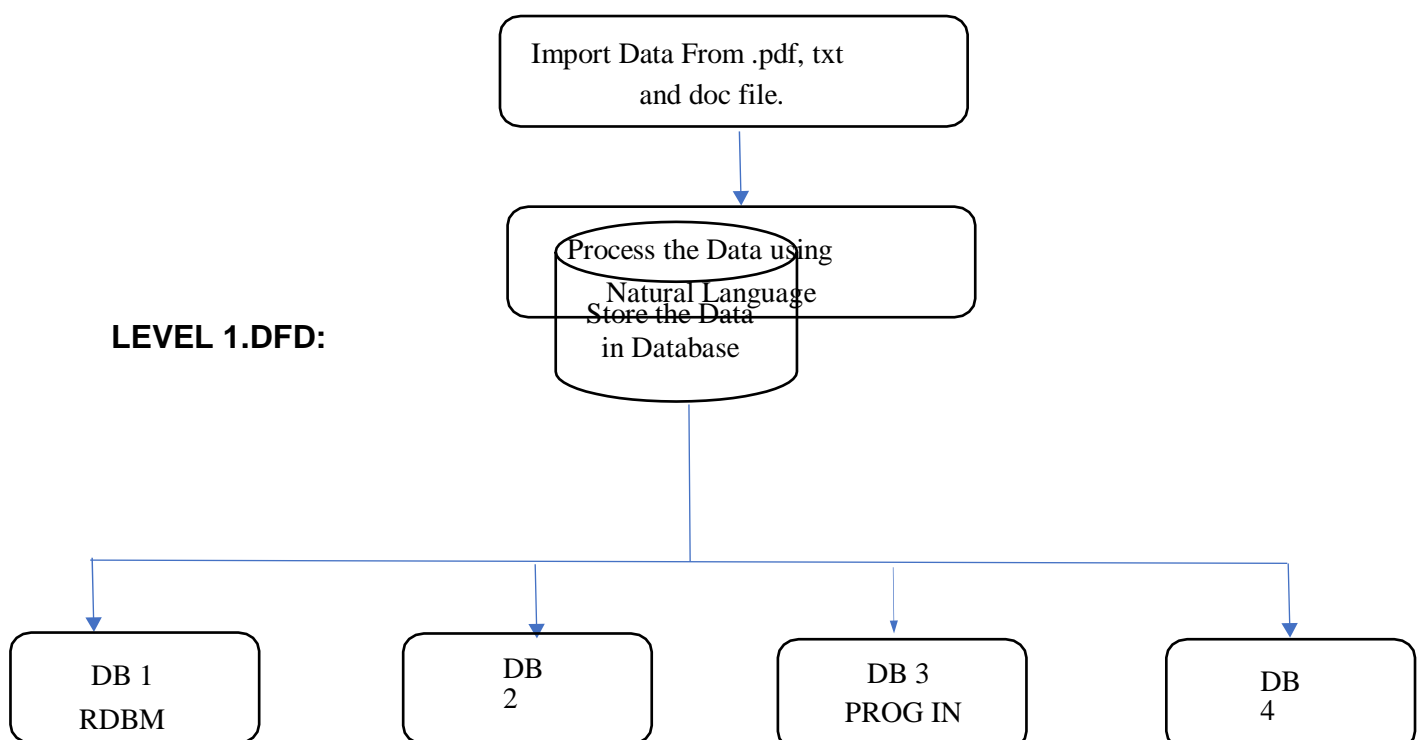
Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. The primary activity during data design is to select logical representation of data object identified. A database file allows to organize data into tables. A table is a subset of entire database. A database contains one or more tables. The use of this database is used to specify the purpose of each name to have

their own datatype and value to be presented for it.

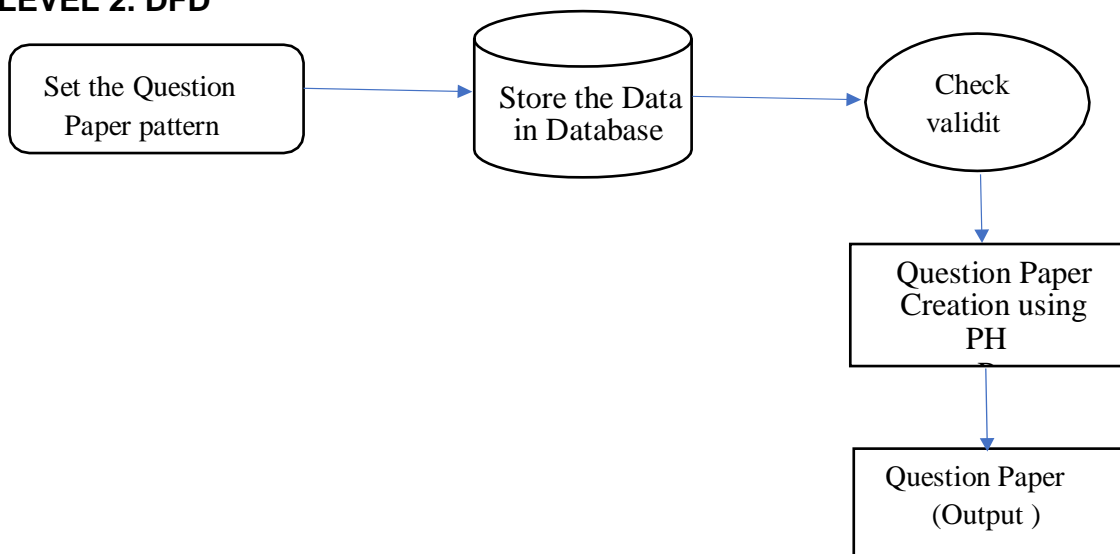
#### 4.1 DATA FLOW DIAGRAM:

The DFD takes an input-process-output view of a system i.e. data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software. Data objects represented by labelled arrows and transformation are represented by circles also called as bubbles. DFD is presented in a hierarchical fashion i.e. the first data flow model represents the system as a whole. Subsequent DFD refine the context diagram (level 0 DFD), providing increasing details with each subsequent level. The DFD enables the software engineer to develop models of the information domain & functional domain at the same time. As the DFD is refined into greater levels of details, the analyst performs an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of the data as it moves through the process that embodies the applications. A context-level DFD for the system the primary external entities produce information for use by the system and consume information generated by the system. The labelled arrow represents data objects or object hierarchy.

##### LEVEL 0.DFD:

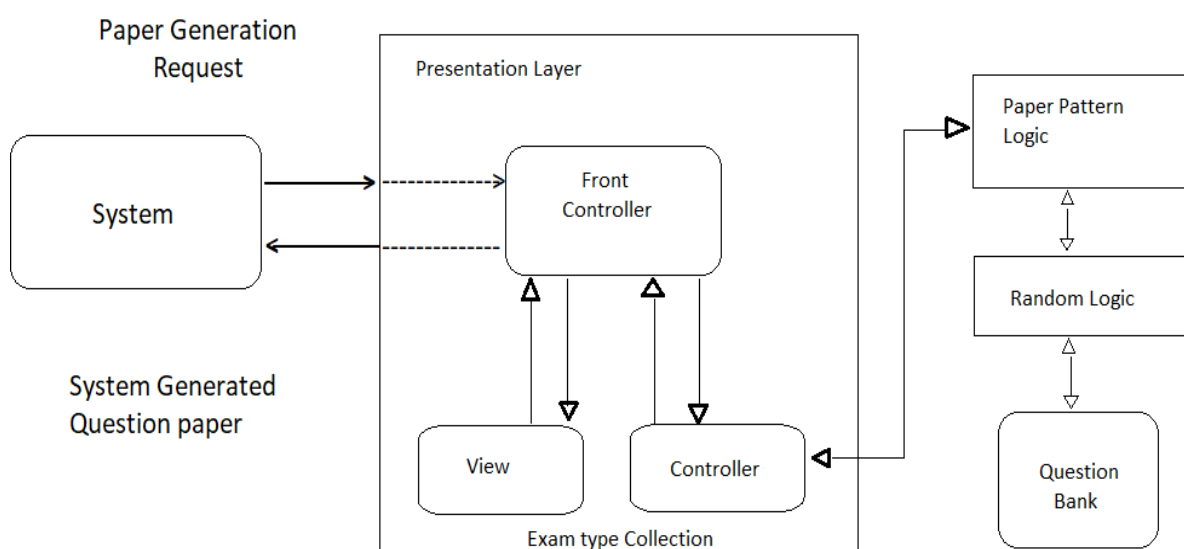


## LEVEL 2. DFD



This Data flow diagram represents the output procedure from the database. Using the above flow diagram, the software produces the output.

## The Overall Data Flow Diagram.



## 4.4 ENTITY RELATIONSHIP DIAGRAM

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 as a way to unify the network and relational database views.

Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity Relationship diagram which is used to visually represent data objects.

Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

### **Connectivity and Cardinality:**

The basic types of connectivity for relations are: one-to-one, one-to-many, and many- to-many. A one-to-one (1:1) relationship is when at most one instance of an Entity A is associated with one instance of entity B. For example, "employees in the company are each assigned their own office. For each employee there exists a unique office and for each office there exists a unique employee.

A one-to-many (1: N) relationships is when for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity A. An example of a 1: N relationships are:

- A department has many employees
- Each employee is assigned to one department

A many to many (M:N) relationship, sometimes called non-specific, is when for one instance of entity A, there are zero, one, or many instances of entity B and for one instance of entity B there are zero, one, or many instance of entity A.

The connectivity of a relationship describes the mapping of associated.

## **ER Notations**

There is no standard for representing data objects in ER diagrams. Each modelling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non- academics. Today, there are a number of notations used, among the more common are Bachman, crow's foot, and IDEFIX. All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection.

The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

- Entities are represented by labelled rectangles. The label is the name of the entity. Entity names should be singular nouns.
- Relationships are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs Attributes, when included, are listed inside the entity rectangle.
- Attributes which are identifiers are underlined. Attribute names should be singular nouns.
- Cardinality of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.
- Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.





Each row in a relational is uniquely identified by a primary key. This can be by one or more sets of column values. In most scenarios it is a single column, such as employee ID.

Every relational table has one primary key. Its purpose is to uniquely identify each row in the database. No two rows can have the same primary key value. The practical result of this is that you can select every single row by just knowing its primary key.

**Question Table 1 (RDBMS):**

Column Name	Data Type	Description
ID	Int (Primary Key)	Identify Data
Section A	Varchar	SectionA Questions
Section B	Varchar	SectionB Questions
Section C	Varchar	SectionC Questions

**Question Table 2 (PROGRAMMING IN JAVA):**

Column Name	Data Type	Description
ID	Int (Primary Key)	Identify Data
Section A	Varchar	Section A Questions
Section B	Varchar	Section B Questions
Section C	Varchar	Section C Questions

**Question Table 3 (PROGRAMMING IN C):**

Column Name	Data Type	Description
ID	Int (Primary Key)	Identify Data
Section A	Varchar	Section A Questions
Section B	Varchar	Section B Questions
Section C	Varchar	Section C Questions

**Question Table 4 (PROGRAMMING IN C++):**

Column Name	Data Type	Description
ID	Int (Primary Key)	Identify Data
Section A	Varchar	Section A Questions
Section B	Varchar	Section B Questions
Section C	Varchar	Section C Questions

#### 4.6. INPUT DESIGN

### Question Paper Automation

Subject Name

Relational Database Management System

Subject Code

19CAU18

Section A

13

Total No Of Questions

10

No Of Question To Be Answered

Section B

8

Total No Of Questions

5

No Of Question To Be Answered

Section C

7

Total No Of Questions

5

No Of Question To Be Answered

Marks

2

Section A

6

Section B

10

Section C

Total Mark

☐ 100

☒ 50

Generate

This is the input design of this project, where we can set pattern of the question paper.

## 4.7 OUTPUT DESIGN:

10/31/21, 10:36 AM

19CAU18 \_ Relational Database Management System

**PSG COLLEGE OF ARTS & SCIENCE**  
**DEPARTMENT OF COMPUTER APPLICATION**  
**COMPREHENSIVE EXAMINATION- June 2021**  
**Relational Database Management System- 19CAU18**

**Max Mark : 100.**  
**Time : 3 hrs.**

**SECTION A (10 X 2 = 20 Marks)**

**Answer any 10 Question**

1. What is Transaction?
2. List the different types of database-system users.
3. What is external level of architecture?
4. Define Data Independence.
5. What is internal level of architecture?
6. Write about the role of Transaction manager.
7. What are the different DBMS languages?
8. What is external level of architecture?
9. List the purpose of Database System.
10. Write the difference between centralized and distributed database.
11. What is conceptual level of architecture?
12. Define: Data Dictionary
13. What are the Basic Components of a DBMS?

**SECTION B (5 X 6 = 30 Marks)**

**Answer any 5 Question**

14. Explain about Application types.
15. Write short notes on data types.
16. What are the components of a DBMS system? Explain
17. Discuss briefly the data entry through multi-table form.
18. Give the purpose of data dictionary.
19. Discuss briefly the desirable properties of decompositions
20. Give the purpose of data dictionary.
21. Discuss briefly about DBMS

**SECTION C (5 X 10 = 50 Marks)**

**Answer any 5 Question**

22. List down the components of DBMS. Explain DBMS with its diagram.
23. What are the Data Integrity constraints?
24. What are Codd's Rules?
25. What is Data Independence?
26. What is Data Independence?
27. Explain about Three Levels of Architecture in DBMS.
28. What are the advantages of a DBMS? Explain them.

\*\*\*\*\*

This is the output design of this project, where we can view and print the question paper

## **5.SYSTEM TESTING AND SECURITY**

### **5.1 SOFTWARE TESTING:**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

#### **Types of testing:**

1. Unit testing
2. White-box testing
3. Black-box testing
4. Validation testing
5. Backend testing

#### **Unit Testing:**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### **White-box Testing:**

It is a test case design method that uses the control structure of the procedural design to drive test cases. Using white box testing methods it was guaranteed that most of the independent paths within modules had been exercised at least once, all logical decision on their true and false sides, executed all loops at their boundaries and exercised internal data structures to ensure their data validity.

White box testing has been done to achieve the following objectives. Logic errors and incorrect assumptions are inversely proportional to the probability that a program path will be executed. Errors tend to creep into the work when design and implementation functions, condition or control that is out of the mainstream. We often believe that logical path is not likely to be executed when the fact it may be executed on regular basis. When program is translated into programming language source code, it is likely that some typing errors will occur. Many will be uncovered by syntax and type checking mechanism but others may go undetected until testing begins.

### **Black-box Testing:**

Although tests are designed to uncover errors, they are also used to demonstrate that the software functions are operational, input is properly accepted and output is correctly produced and that the integrity of external information is maintained. A black box test examines some of fundamental aspects of a system with little regard for the internal logical structure of the software.

All input screens were thoroughly tested for data validity and smoothness of data entry operations. Test cases were so formulated to verify whether the system works properly in rare conditions also. Error conditions were checked. Data entry operations are to be user friendly and smooth. It would be easier for the operators if they can enter data through key board only. Using mouse and keyboard alternatively will affect the speed of data entry. Care was taken to make data entry as smooth as possible. Flow of object was made convenient to the data entry operations. Testing also tested wrong input for data value.

### **Validation Testing:**

Validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer. After validation tests have been conducted one of the two possible conditions exists.

The function or performance characteristics are acceptable and confirmed to specification. A decision from specification is uncovered and defining list is created. System validation checks the quality of software in both simulated and live environment. First the software goes through a phase in which errors and failures based on simulated user requirements are verified and studied.

### **Back-end Testing:**

Whenever an input or data is entered on front-end application, it stores in the database and the testing of such database is known as Database Testing or Backend testing. There are different databases like SQL Server, MySQL. Database testing involves testing of table structure, schema, stored procedure, data structure and so on.

### **Functional testing:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

### **Functional testing is centered on the following items:**

1. Valid Input: identified classes of valid input must be accepted.
2. Invalid Input: identified classes of invalid input must be rejected.
3. Functions: identified functions must be exercised.
4. Output: identified classes of application outputs must be exercised.

## 5.2 SECURITY

Software security is an idea implemented to protect software against malicious attack and other hacker risks so that the software continues to function correctly under such potential risks. Security is necessary to provide integrity, authentication and availability.

Any compromise to integrity, authentication and availability makes a software insecure. Software systems can be attacked to steal information, monitor content, introduce vulnerabilities and damage the behavior of software. Malware can cause DoS (denial of service) or crash the system itself.

Buffer overflow, stack overflow, command injection and SQL injections are the most common attacks on the software.

Buffer and stack overflow attacks overwrite the contents of the heap or stack respectively by writing extra bytes.

Command injection can be achieved on the software code when system commands are used predominantly. New system commands are appended to existing commands by the malicious attack. Sometimes system command may stop services and cause DoS.

SQL injections use malicious SQL code to retrieve or modify important information from database servers. SQL injections can be used to bypass login credentials. Sometimes SQL injections fetch important information from a database or delete all important data from a database.

The only way to avoid such attacks is to practice good programming techniques. System-level security can be provided using better firewalls. Using intrusion detection and prevention can also aid in stopping attackers from easy access to the system.



## **6.SCOPE FOR FUTURE ENHANCEMENT**

This software can be widely used in educational institutes etc. It is a web Based application so user location doesn't matter. Admin can access this software anywhere anytime. This system also evaluates the candidate's capability and skills efficiently.

It is a fully automated system which results in fast results. In this system there is no need of transporting paper through police/security vans to all colleges. This system provides unbiased results. The usage of this system reduces human effort and saves time and resources to an extent.

This project can be further extended to generate yes/no type questions and fill in the blanks type questions. Further automatic answer copy checking systems can be added for checking the answer sheets and providing the result to the student

## **7. CONCLUSION**

The question selection difficulty has been formulated as a multi-constraint optimization issue, which aims to produce question papers that meet several constraints stated by the paper setter. Intelligent Question Paper Generator will generate a well formatted question paper in a matter of a few seconds i.e. saving a lot of time when compared to traditional systems.

With the use of this system for question paper generation there are no chances of exam paper getting 5 leaked as paper is generated just before the test. It will save a lot of time for teachers and thus will improve efficiency. The implemented process strives to solve the problems listed above in a positive way. The implemented work narrates an automated system that heads away from the traditional process of paper generation to an automated process, by giving controlled entry to the resources that is attained by involving users and their roles in the colleges.

We have also considered the importance of randomization in the process of paper generation. This system uses a logical algorithm which is absolutely randomized in

nature and it also avoids duplication of questions. And therefore, the resultant automated system for Question Paper Generation will yield a growth in phrase of random creation of question papers and also a secured platform

## **8. BIBILOGRAPHY**

### **BOOKS REFERENCES:**

1. Himanshu Jethwani, Mohd Shahid Husain, Mohd Akbar, ISSN 2347 - 7911 "International Journal for Innovations in Engineering, Science and Management" Available online at: [www.ijiesm.com](http://www.ijiesm.com) Volume 3, Issue 4, April 2015
2. Isha Sood, 7 Characteristics Of A Good Question. Online available at: <https://elearningindustry.com/characteristics-of-a-good-question-7>
3. Paul DuBois, "MySQL Developer's Library", 4th ed, 2009 (5th ed is probably available).
4. Codd E. F., "A Relational Model of Data for Large Shared Data Banks", Communications of the ACM, vol. 13, issue 6, pp. 377–387, June 1970.  
@<http://download.oracle.com/javase/tutorial/jdbc/basics/index.html>.
5. Russell Dyer, "MySQL in a Nutshell", 2nd ed, O'Reilly, 2008.
6. The Joy of PHP Programming: A Beginner's Guide – by Alan Forbes
7. PHP & MySQL Novice to Ninja – by Kevin Yank

### **WEBSITES**

1. W3School HTML/CSS Tutorials, References and Examples @ <http://www.w3schools.com/>. (W3School is not

related to W3C).

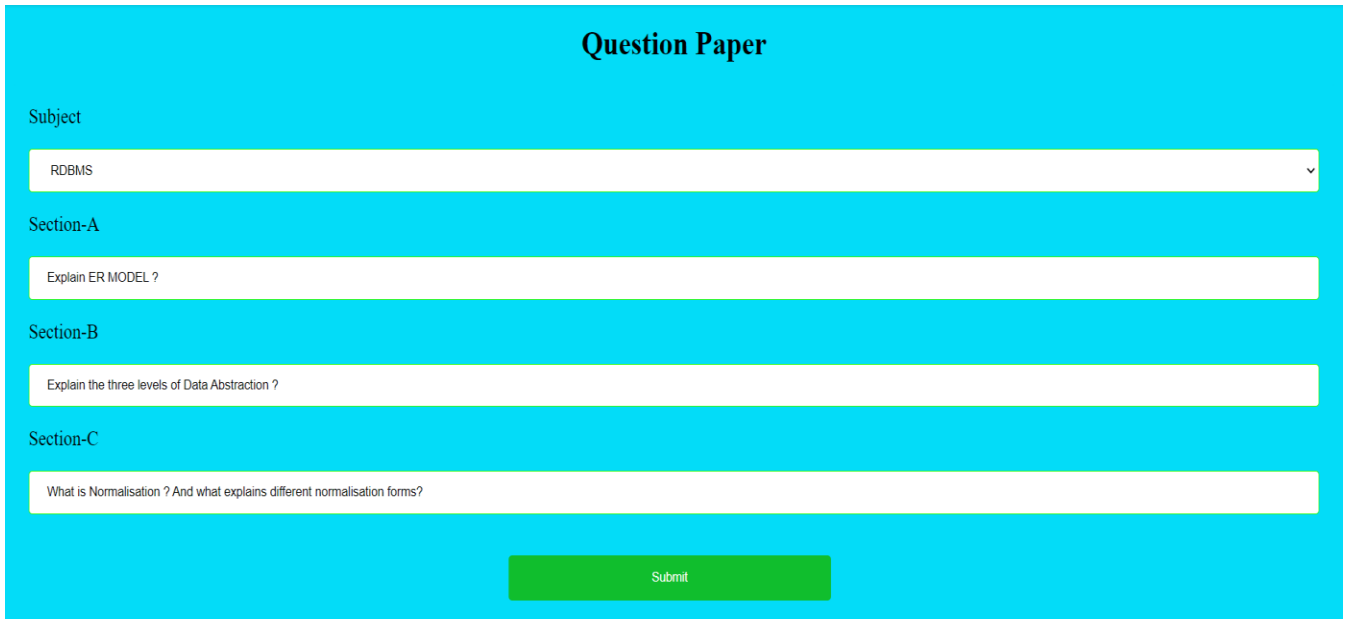
2. Mozilla's (MDN) "JavaScript Guide" @  
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>.
3. W3School JavaScript Tutorials, References and PHP mother site  
@<http://php.net/>.
4. PHP Manual @ <http://php.net/manual/en/>.
5. PHP Language Reference @ <http://php.net/manual/en/langref.php>.
6. MySQL Mother Site @[www.mysql.com](http://www.mysql.com).
7. MySQL 5.7 "Reference Manual" @<http://dev.mysql.com/doc/>.
8. MySQL 5.7 "SQL Statement Syntax"
9. MySQL 5.7 "SQL Statement Syntax"

<http://dev.mysql.com/doc/refman/5.7/en/sql-syntax.html>

## 9. APPENDIX

### 10.1. SCREENSHOTS

#### Import Questions Page:



The screenshot shows a web form titled "Question Paper" with a light blue background. The form contains the following elements:

- Subject:** A dropdown menu with "RDBMS" selected.
- Section-A:** A text input field containing "Explain ER MODEL ?".
- Section-B:** A text input field containing "Explain the three levels of Data Abstraction ?".
- Section-C:** A text input field containing "What is Normalisation ? And what explains different normalisation forms?".
- Submit:** A green button at the bottom center.

This page is the first module, Importing question module. Here we can enter the questions and it will be stored in the Question Bank (Database).

### Pattern Setting Page:

## Question Paper Automation

Subject Name

Relational Database Management System

Subject Code

19CAU18

Section A

13

Total No Of Questions

10

No Of Question To Be Answered

Section B

8

Total No Of Questions

5

No Of Question To Be Answered

Section C

7

Total No Of Questions

5

No Of Question To Be Answered

Marks

2

Section A

6

Section B

10

Section C

Total Mark

☐ 100

☒ 50

Generate

This is the Input Design, Pattern Setting Module. Here we can set the question paper pattern

## Output Page:

10/31/21, 10:36 AM

19CAU18 \_ Relational Database Management System

**PSG COLLEGE OF ARTS & SCIENCE**  
**DEPARTMENT OF COMPUTER APPLICATION**  
**COMPREHENSIVE EXAMINATION- June 2021**  
**Relational Database Management System- 19CAU18**

**Max Mark : 100.**  
**Time : 3 hrs.**

**SECTION A (10 X 2 = 20 Marks)**

**Answer any 10 Question**

1. What is Transaction?
2. List the different types of database-system users.
3. What is external level of architecture?
4. Define Data Independence.
5. What is internal level of architecture?
6. Write about the role of Transaction manager.
7. What are the different DBMS languages?
8. What is external level of architecture?
9. List the purpose of Database System.
10. Write the difference between centralized and distributed database.
11. What is conceptual level of architecture?
12. Define: Data Dictionary
13. What are the Basic Components of a DBMS?

**SECTION B (5 X 6 = 30 Marks)**

**Answer any 5 Question**

14. Explain about Application types.
15. Write short notes on data types.
16. What are the components of a DBMS system? Explain
17. Discuss briefly the data entry through multi-table form.
18. Give the purpose of data dictionary.
19. Discuss briefly the desirable properties of decompositions
20. Give the purpose of data dictionary.
21. Discuss briefly about DBMS

**SECTION C (5 X 10 = 50 Marks)**

**Answer any 5 Question**

22. List down the components of DBMS. Explain DBMS with its diagram.
23. What are the Data Integrity constraints?
24. What are Codd's Rules?
25. What is Data Independence?
26. What is Data Independence?
27. Explain about Three Levels of Architecture in DBMS.
28. What are the advantages of a DBMS? Explain them.

\*\*\*\*\*

This is the Output Design, Question paper viewing module. Here we can view and print the question paper.

## 10.2. SOURCE CODE:

### Import Question.html

```
<!DOCTYPE html>
<html>
<style>
  body {
    background-color: #03dcf8;
  }
  h1 {text-align:
center;} label {font-
size: 20px;}
label {font-family: "Times New Roman", Times,
serif;} input[type=text], select {
  width: 100%;
  padding: 12px
20px; margin: 20px
0; display: inline-
block;
border: 1px solid rgb(45, 248,
18); border-radius: 4px;
box-sizing: border-box;
}

input[type=submit]
{ width: 25%;
background-color:
#10be2d; color: rgb(255,
255, 255); padding: 14px
20px; margin: 20px
550px; border: none;
```

`border-radius: 4px;`



```
    cursor: pointer;
}
```

```
input[type=submit]:hover {
    background-color:
    #0cd2e0;
}
```

```
div {
    border-radius: 5px;
    background-color:
    #03dcf8; padding: 20px;
}
```

```
</style>
```

```
<body>
```

```
<h1>Question Paper</h1>
```

```
<div>
```

```
    <form action="import.php" method="POST">
```

```
        <label for="Question_Paper">Subject</label>
```

```
        <select id="Subject" name="Question_Paper">
```

```
            <option value="null">Select Subject</option>
```

```
            <option value="rdbms">RDBMS</option>
```

```
            <option value="c_prog">C Program</option>
```

```
            <option value="c_plusplus">C++ Program</option>
```

```
            <option value="java">JAVA</option>
```

```
        </select>
```

```
        <label for="sa">Section-A</label>
```

```
        <input type="text" id="sa" name="sectiona" placeholder="Enter the
Section- A Question...">
```

```

<label for="sb">Section-B</label>
<input type="text" id="sb" name="sectionb" placeholder="Enter the
Section- B Question...">

<label for="sc">Section-C</label>
<input type="text" id="sc" name="sectionc" placeholder="Enter the
Section- C Question...">
<input type="submit" value="Submit">
</form>
</div>

</body>
</html>

```

### **Question\_Generator.html**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Question Paper Generator</title>
  <style>
    *{
      margin:0;
      padding:
      0;
    }
    body{

      background-image:
      url(logo.png); margin-
      top:40px;

```

background-position:center;

```
background-repeat : no-  
repeat;      background-  
size:300px 300px; font-  
family:sans-serif;  
}
```

```
.regform{  
    width:800px;  
background-  
color:rgb(0,0,0,0.6);  
margin:auto;  
color:#FFFFFF;  
padding:10px 0px 10px  
0px; text-align:center;  
border-radius:15px 15px 0px 0px ;  
}
```

```
.main{  
    background-  
color:rgb(0,0,0,0.7);  
width:800px;  
margin:auto;}
```

```
form{  
    padding:10px;  
  
}
```

```
#name{  
    width:100%  
;  
height:100px  
;
```

```
}  
.name{  
    margin-left:25px;
```

```

margin-
top:30px; width:
110px;
    color: white;
    font-size:
    16px;
    font-weight: 700;}
.firstname{
    position:
    relative; left:150px;
    top:-37px;
    line-height: 40px;
    border-radius:
    6px; padding: 0
    22px; font-size:
    16px;

}
.lastname{
    position:
    relative; left:417px;
    top:-80px;
    line-height: 40px;
    border-radius:
    6px; padding: 0
    22px; font-size:
    16px; color:#555;
}
.firstlabel{
    position:relativ
    e;
    color:#E5E5E5

```

```
;  
text-transform:  
capitalize; font-size:  
14px;  
left:150px;
```

```

top:-45px;
}
.lastlabel{
    position:relativ
e; color:#E5E5E5;
text-
transform:capitalize;
font-size:14px;
left:480px;
top:-76px;
}
.company{
    position:relativ
e; left:150px;
top:-37px;
line-height:
40px;
width:450px;
border-radius:
6px; padding: 0
22px; font-size:
16px; color: #555;
}
.email{
    position:relativ
e; left:200px;
top:-37px;
line-height: 40px;
width:480px;
border-radius:
6px; padding: 0
22px; font-size:

```



```
16px; color: #555;  
}
```

```
.Code{
    position:relativ
e; left:150px;
    top:-37px;
    line-height: 40px;
    width:95px;
    border-radius:
6px; padding: 0
22px; font-size:
16px; color: #555;
}
```

```
.number{
    position:relativ
e; left:140px;
    top:-37px;
    line-height: 40px;
    width:95px;
    border-radius:
6px; padding: 0
22px; font-size:
16px; color: #555;
}
```

```
.area-code{
    position:relativ
e; color:#E5E5E5;
    text-
transform:capitalize;
    font-size:16px;
    left:10px;
    top:-2px;
```

```

    }
.phone-number{
    position:relativ
    e; color:#E5E5E5;
    text-
    transform:capitalize;
    font-size:16px;
    left:-
    10px;
    top:-2px;
}
.option{
    position:relativ
e; left:150px;
    top:-30px;
    line-height:
    40px;
    width:500px;
    height:40px;
    border-radius:
    6px; padding: 0
    22px; font-size:
    16px; color: #555;
    outline:none;
    overflow:hidden;
}
.option option{
    font-size:20px;
}
#coustomer{
    margin-
    left:25px; color:white;

```

font-size:18px;

```

    }
    .radio{
        display:inline-
block; padding-
right:70px; font-
size:25px;
margin-
left:25px;
margin-
top:15px;
color:white;
    }
    .radio input{
        width:20px;
        height:20px;
        border-
radius:50%;
        cursor:pointer;
        outline:none;
    }
    button{
        background-
color:#3BAF9F;
        display:block;
        margin:20px 0px 0px
20px; text-align:center;
        border-radius:12px;
        border:2px solid
#366473; padding:14px
110px; outline:none;
        color:white;
        cursor:pointer;

```

```
transition:0.25p
x;
}
button:hover{
    background-color:#5390F5;
```

```

    }
</style>
</head>
<body>
    <div class="regform">
        <h1>Question Paper Automation</h1>
    </div>
    <div class="main">

        <form method="POST" action ="output.php">
            <h2
class="name"> Subject
Name</h2>
<select class="option" name="subject">
            <option disabled="disabled" selected="selected">--Choose option--</option>
            <option>Relational Database Management System</option>
            <option>Programming in C</option>
            <option>Programming in C++</option>
            <option>Programming in Java</option>
        </select>
        <h2
class="name">
Subject Code </h2>
<input class="company" type="text" name="SubCode">

        <h2
class="name">
Section A</h2>
<input class="Code" type="number" name="SecALim">
        <label class="area-code">Total No of Questions</label>
        <input class="number" type="number" name="SecAAtt">
        <label class="phone-number">No of question to be answered</label>

```

```
<h2
class="name"> Section
B</h2>
```



```

<input class="Code" type="number" name="SecBLim">
    <label class="area-code">Total No of Questions</label>
    <input class="number" type="number" name="SecBAtt">
    <label class="phone-number">No of question to be answered</label>

    <h2
class="name"> Section
C</h2>
<input class="Code" type="number" name="SecCLim">
    <label class="area-code">Total No of Questions</label>
    <input class="number" type="number" name="SecCAtt">
    <label class="phone-number">No of question to be answered</label>

    <h2
class="name">
Marks</h2>
<input class="Code" type="number" name="SecAMark">
    <label class="area-code">Section A</label>
    <input class="number" type="number" name="SecBMark">
    <label class="phone-number">Section B</label>
    <input class="number" type="number" name="SecCMark">
    <label class="phone-number">Section C</label>
    <h2
id="coustomer"> Total
Mark</h2>
<label class="radio">
    <input class="radio-
one" type="radio" checked="checked" name="MaxMark" value = "100">
    <span
class="checkmark"></span>
    100
</label>

```

```
<label class="radio">
```

```
<input class="radio-two" type="radio" name="MaxMark" value = "50">
```

```

        <span
            class="checkmark"></span> 50
    </label>

    <button type="submit">Generate</button>

```

```

    </form>
</div>
</body>
</html>

```

### **Output.php**

```

<?php
$SecALim = $_POST['SecALim'];
$SecBLim = $_POST['SecBLim'];
$SecCLim = $_POST['SecCLim'];
$Sub_Name = $_POST['subject'];
$Sub_Code = $_POST['SubCode'];
$SecAAtt = $_POST['SecAAtt'];
$SecBAtt = $_POST['SecBAtt'];
$SecCAtt = $_POST['SecCAtt'];
$SecAMark = $_POST['SecAMark'];
$SecBMark = $_POST['SecBMark'];
$SecCMark = $_POST['SecCMark'];
$MaxMark =
$_POST['MaxMark'];
if($MaxMark==100)
    $Time =
3; else
    $Time =
2; echo

```

"<html>

```

<head><title>$Sub_Code _ $Sub_Name</title>
</head>
</html>";
echo "<CENTER><b>PSG College Of Arts And
Science</b></CENTER>"; echo "<CENTER><b>June
2021</b></CENTER>";
echo "<CENTER><b>",$Sub_Name,</b></CENTER>";
echo "<CENTER><b> Subject Code -
",$Sub_Code,</b></CENTER><br>"; echo "<div align ='right'>";
echo "<b>Max Mark :
",$MaxMark,</b><br>"; echo "<b> Time :
",$Time," hrs.</b></div>";
if($Sub_Name == "Relational Database Management System")
{
    $host='localhost';
    $username='root';
    $password="";
    $conn=mysqli_connect($host,$username,$password,"question_
bank"); if(!$conn){
        die('Could not Connect My Sql:' .mysql_error());
    }
    else
    {
        echo "<CENTER><b>SECTION A (",$SecAAtt," X ",$SecAMark," =
",$SecAAtt*$SecAMark," Marks)</b></CENTER>";
        echo "<b>Answer any $SecAAtt Question</b><br>";
        $query = "SELECT Section_A FROM `question_bank`.`rdbsms` ORDER BY
RAND() li mit $SecALim";
        $result =
        mysqli_query($conn,$query);
        if($result = $conn->query($query))
        {

```

```
$i = 1;  
while ($row = $result->fetch_assoc())
```

```

        {
            $field1name =
            $row["Section_A"]; echo
            "<br>",$i, ". ", $field1name;
            $i=$i+1;
        }
    }
}

if(!$conn){
    die('Could not Connect My Sql:' .mysql_error());
}
else
{
    echo "<CENTER><b>SECTION B (",$SecBAtt," X ",$SecBMark," =
", $SecBAtt*$Sec BMark," Marks)</b></CENTER>";
    echo "<b>Answer any $SecBAtt Question</b><br>";
    $query1 = "SELECT Section_B FROM `question_bank`.`rdbms` ORDER BY
RAND() limit $SecBLim";
    $result1 =
    mysqli_query($conn,$query1);
    if($result1 = $conn->query($query1))
    {
        while ($row1 = $result1->fetch_assoc())
        {
            $field2name =
            $row1["Section_B"]; echo
            "<br>",$i, ". ", $field2name;
            $i=$i+1;
        }
    }
}

if(!$conn){

```

```
die('Could not Connect My Sql:' .mysql_error());  
}
```



```

else
{
    echo "<CENTER><b>SECTION C (",$SecCAtt," X ",$SecCMark," =
",,$SecCAtt*$SecCMark," Marks)</b></CENTER>";
    echo"<b>Answer any $SecCAtt Question</b><br>";
    $query2 = "SELECT Section_C FROM `question_bank`.`rdbms` ORDER BY
RAND() limit $SecCLim";
    $result2 =
    mysqli_query($conn,$query2);
    if($result2 = $conn->query($query2))
    {
        while ($row2 = $result2->fetch_assoc())
        {
            $field3name =
            $row2["Section_C"]; echo
            "<br>",$i,". ",$field3name;
            $i=$i+1;
        }
    }
    echo "<center><br><br>*****</center>";
}
}
elseif($Sub_Name == "Visual Basics")
{
    $host2='localhost';
    $username2='root';
    $password2="";
    $conn2=mysqli_connect($host2,$username2,$password2,"question
_bank"); if(!$conn2){
        die('Could not Connect My Sql:' .mysql_error());
    }
}
else

```

{

```

        echo "<CENTER><b>SECTION A (",$SecAAtt," X ",$SecAMark," =
",,$SecAAtt*$Sec cAMark," Marks)</b></CENTER>";

        echo"<b>Answer any $SecAAtt Question</b><br>";

        $query2 = "SELECT Section_A FROM question_bank.c_prog ORDER BY
RAND() li mit $SecALim";

        $result2 =
        mysqli_query($conn2,$query2);
        if($result2 = $conn2->query($query2))
        {
            $i = 1;
            while ($row2 = $result2->fetch_assoc())
            {
                $field1name2 =
                $row2["Section_A"]; echo
                "<br>",$i,". ",$field1name2;
                $i=$i+1;
            }
        }
    }
    if(!$conn2){
        die('Could not Connect My Sql!' .mysql_error());
    }
    else
    {
        echo "<CENTER><b>SECTION B (",$SecBAtt," X ",$SecBMark," =
",,$SecBAtt*$Sec BMark," Marks)</b></CENTER>";

        echo"<b>Answer any $SecBAtt Question</b><br>";

        $query12 = "SELECT Section_B FROM question_bank.c_prog ORDER BY
RAND() li mit $SecBLim";

        $result12 =
        mysqli_query($conn2,$query12);
        if($result12 = $conn2->query($query12))

```

```
{  
    while ($row12 = $result12->fetch_assoc())  
    {
```

```

        $field2name2 =
        $row12["Section_B"]; echo
        "<br>",$i,". ",$field2name2;
        $i=$i+1;
    }
}
}
if(!$conn2){
    die('Could not Connect My Sql:' .mysql_error());
}
else
{
    echo "<CENTER><b>SECTION C (",$SecCAtt," X ",$SecCMark," =
",$SecCAtt*$SecCMark," Marks)</b></CENTER>";
    echo"<b>Answer any $SecCAtt Question</b><br>";
    $query22 = "SELECT Section_C FROM question_bank.c_prog ORDER BY
RAND() li mit $SecCLim";
    $result22 =
    mysqli_query($conn2,$query22);
    if($result22 = $conn2-
    >query($query22)){
        while ($row22 = $result22->fetch_assoc())
        {
            $field3name2 =
            $row22["Section_C"]; echo
            "<br>",$i,". ",$field3name2;
            $i=$i+1;
        }
    }
    echo "<center><br><br>*****</center>";
}
}

```

```
elseif($Sub_Name == "Programming in C")  
{  
    $host3='localhost';
```

```

$username3='root';
$password3="";
$conn3=mysqli_connect($host3,$username3,$password3,"question
_bank"); if(!$conn3){
die('Could not Connect My Sql:' .mysql_error());
}
else
{
echo "<CENTER><b>SECTION A (",$SecAAtt," X ",$SecAMark," =
",$SecAAtt*$SecAMark," Marks)</b></CENTER>";
echo"<b>Answer any $SecAAtt Question</b><br>";
$query3 = "SELECT Section_A FROM `question_bank`.`c_plusplus` ORDER BY
RAND() limit $SecALim";
$result3 =
mysqli_query($conn3,$query3);
if($result3 = $conn3->query($query3))
    $i = 1;
    while ($row3 = $result3->fetch_assoc())
    {
        $field1name3 =
        $row3["Section_A"]; echo
        "<br>",$i,". ",$field1name3;
        $i=$i+1;
    }
}
if(!$conn3){
die('Could not Connect My Sql:' .mysql_error());
}
else
{
echo "<CENTER><b>SECTION B (",$SecBAtt," X ",$SecBMark," =
",$SecBAtt*$SecBMark," Marks)</b></CENTER>";

```

echo"<b>Answer any \$SecBAtt Question</b><br>";



```

$query13 = "SELECT Section_B FROM `question_bank`.`c_plusplus` ORDER BY
RAND() limit $SecBLim";
$result13 =
mysqli_query($conn3,$query13);
if($result13 = $conn3->query($query13))
    while ($row13 = $result13->fetch_assoc())
    {
        $field2name3 =
        $row13["Section_B"]; echo
        "<br>",$i, ". ", $field2name3;
        $i=$i+1;
    }
}
if(!$conn3){
    die('Could not Connect My Sql:' .mysql_error());
}
else
{
    echo "<CENTER><b>SECTION C (",$SecCAtt," X ",$SecCMark," =
    ",$SecCAtt*$SecCMark," Marks)</b></CENTER>";
    echo"<b>Answer any $SecCAtt Question</b><br>";
    $query23 = "SELECT Section_C FROM `question_bank`.`c_plusplus` ORDER BY
    RAND() limit $SecCLim";
    $result23 =
    mysqli_query($conn3,$query23);
    if($result23 = $conn3->query($query23))
        while ($row23 = $result23->fetch_assoc())
        {
            $field3name3 =
            $row23["Section_C"]; echo
            "<br>",$i, ". ", $field3name3;
            $i=$i+1;
        }
    }
}

```

```
}  
echo "<center><br><br>*****</center>";  
}  
}
```

```

elseif($Sub_Name == "Programming in C++")
{
    $host4='localhost';
    $username4='root';
    $password4="";
    $conn4=mysqli_connect($host4,$username4,$password4,"question
_bank"); if(!$conn4){
    die('Could not Connect My Sql:' .mysql_error());
}
else
{
    echo "<CENTER><b>SECTION A (",$SecAAtt," X ",$SecAMark," =
    ",$SecAAtt*$SecAMark," Marks)</b></CENTER>";
    echo"<b>Answer any $SecAAtt Question</b><br>";
    $query4 = "SELECT Section_A FROM `question_bank`. `java` ORDER BY RAND()
    limit $ SecALim";
    $result4 =
    mysqli_query($conn4,$query4);
    if($result4 = $conn4->query($query4))
        $i = 1;
        while ($row4 = $result4->fetch_assoc())
        {
            $field1name4 =
            $row4["Section_A"]; echo
            "<br>",$i, ". ",$field1name4;
            $i=$i+1;
        }
    }
    if(!$conn4){
        die('Could not Connect My Sql:' .mysql_error());
    }
    else

```

{

```

echo "<CENTER><b>SECTION B (",$SecBAtt," X ",$SecBMark," =
",$SecBAtt*$SecBMark," Marks)</b></CENTER>";
echo"<b>Answer any $SecBAtt Question</b><br>";
$query14 = "SELECT Section_B FROM `question_bank`.`java` ORDER BY RAND() limit
$SecBLim";
$result14 =
mysqli_query($conn4,$query14);
if($result14 = $conn4->query($query14))
    while ($row14 = $result14->fetch_assoc())
    {
        $field2name4 =
        $row14["Section_B"]; echo
        "<br>",$i,". ",$field2name4;
        $i=$i+1;
    }
}
if(!$conn4){
    die('Could not Connect My Sql:' .mysql_error());
}
else
{
    echo "<CENTER><b>SECTION C (",$SecCAtt," X ",$SecCMark," =
    ",$SecCAtt*$SecCMark," Marks)</b></CENTER>";
    echo"<b>Answer any $SecCAtt Question</b><br>";
    $query24 = "SELECT Section_C FROM `question_bank`.`java` ORDER BY RAND() limit
    $SecCLim";
    $result24 =
    mysqli_query($conn4,$query24);
    if($result24 = $conn4->query($query24))
        while ($row24 = $result24->fetch_assoc())
        {
            $field3name4 =

```

```
$row24["Section_C"]; echo  
"<br>",$i,". ",$field3name4;  
$i=$i+1;  
}
```

```

echo "<center><br><br>*****</center>";
}
}
else
{
    $host5='localhost';
    $username5='root';
    $password5='';
    $conn5=mysqli_connect($host5,$username5,$password5,"question
_bank"); if(!$conn5){
    die('Could not Connect My Sql:' .mysql_error());
}
else
{
    echo "<CENTER><b>SECTION A (",$SecAAtt," X ",$SecAMark," =
    ",$SecAAtt*$SecAMark," Marks)</b></CENTER>";
    echo"<b>Answer any $SecAAtt Question</b><br>";
    $query5 = "SELECT Section_A FROM `question_bank`.`question_5` ORDER BY
    RAND() limit $SecALim";
    $result5 =
    mysqli_query($conn5,$query5);
    if($result5 = $conn5->query($query5))
        $i = 1;
        while ($row5 = $result5->fetch_assoc())
        {
            $field1name5 =
            $row5["Section_A"]; echo
            "<br>",$i,". ",$field1name5;
            $i=$i+1;
        }
    }
    if(!$conn5){

```

```
die('Could not Connect My Sql:' .mysql_error());
```



```

}
else
{
echo "<CENTER><b>SECTION B (",$SecBAtt," X ",$SecBMark," =
",$SecBAtt*$SecBMark," Marks)</b></CENTER>";
echo"<b>Answer any $SecBAtt Question</b><br>";
$query15 = "SELECT Section_B FROM `question_bank`.`question_5` ORDER BY
RAND() limit $SecBLim";
$result15 =
mysqli_query($conn5,$query15);
if($result15 = $conn5->query($query15))
    while ($row15 = $result15->fetch_assoc())
    {
        $field2name5 =
        $row15["Section_B"]; echo
        "<br>",$i,". ",$field2name5;
        $i=$i+1;
    }
}
if(!$conn5){
    die('Could not Connect My Sql:' .mysql_error());
}
else
{
echo "<CENTER><b>SECTION C (",$SecCAtt," X ",$SecCMark," =
",$SecCAtt*$SecCMark," Marks)</b></CENTER>";
echo"<b>Answer any $SecCAtt Question</b><br>";
$query25 = "SELECT Section_C FROM `question_bank`.`question_5` ORDER BY
RAND() limit $SecCLim";
$result25 =
mysqli_query($conn5,$query25);
if($result25 = $conn5->query($query25))

```

```
while ($row25 = $result25->fetch_assoc())  
{  
    $field3name5 = $row25["Section_C"];
```

```

        echo "<br>",$i,". ",$field3name5;
        $i=$i+1;
    }
    echo "<center><br><br>*****</center>";
}
}
?>

```

### **Config.PHP**

```

<?php
define('DBSERVER','localhost');
define('DBUSERNAME','root');
; define('DBPASSWORD','');
define('DBNAME','question_bank');
// Create connection
$conn = mysqli_connect(DBSERVER, DBUSERNAME ,DBPASSWORD , DBNAME);
// Check
connection if
(!$conn) {
    die("Connection failed: " . mysqli_connect_error());
} else {
    echo "";
}
?>

```



10. PHP mother site @<http://php.net/>.
11. PHP Manual @ <http://php.net/manual/en/>.
12. PHP Language Reference @ <http://php.net/manual/en/langref.php>.
13. MySQL Mother Site @[www.mysql.com](http://www.mysql.com).
14. MySQL 5.7 "Reference Manual" @<http://dev.mysql.com/doc/>.
15. MySQL 5.7 "SQL Statement Syntax"  
@ <http://dev.mysql.com/doc/refman/5.7/en/sql-syntax.html>

.

