

## **1. ABSTRACT**

## 1.1 ABSTRACT

Our project is the Automatic Question paper generation system which will enable college to automatically produce question papers in the database. The system will have the capacity to process different unique arrangements of papers automatically. It assumes control over the whole dreary undertaking and does the manual, laborious work quickly and efficiently. You can organize the question bank as per subject, class, and section and weightage. The question paper generator is the one and programming for setting examination question paper. The software performs out all tasks identified with the paper setting, starting from preparing question bank to printing paper. Programming is extremely helpful for small medium and large-scale establishments. The software will enable the college with a great instrument to create question papers in a limited capacity to focus time along these lines sparing a considerable measure of their valuable time. The authorities have the flexibility to produce class tests, unit tests, terminal tests, and last tests.

## **2. INTRODUCTION**

## **2.1 AIM AND OBJECTIVE**

**“EASY EXAM”** is an Automatic Question paper generation system which will enable college to automatically produce question papers in the database. The system will have the capacity to process different unique arrangements of papers automatically.

## **2.2SCOPE OF PROJECT**

The software **“EASY EXAM”** is an Automatic Question paper generator system which will enable college to automatically produce question papers in the database. The system will have the capacity to process different unique arrangements of papers automatically. It assumes control over the whole dreary undertaking and does the manual, laborious work quickly and efficiently. You can organize the question bank as per subject, class, and section and weightage. The question paper generator is the one and programming for setting examination question paper. The software performs out all tasks identified with the paper setting, starting from preparing question bank to printing paper. Programming is extremely helpful for small medium and large-scale establishments. The software will enable the college with a great instrument to create question papers in a limited capacity to focus time along these lines sparing a considerable measure of their valuable time. The authorities have the flexibility to produce class tests, unit tests, terminal tests, and last tests.

### **3. SYSTEM STUDY**

### 3.1 EXISTING SYSTEM

In this section, we will discuss the existing system and some of their drawbacks which force us to plan this whole idea of developing Question Paper Generator. In the existing system the Question paper is done by online but it is not in a perfect pattern. In the new application **Easy Exam** is a perfect Question paper generator.

### 3.2 DRAWBACKS OF EXISTING SYSTEM

Drawbacks of the current system are,

- Lack of security of data.
- Time consuming.
- Consumes large volume of paper work.
- Needs manual calculations.

### 3.3 PROPOSED SYSTEM

The aim of proposed system is to develop a system of improved facilities. The Proposed system can overcome all the imitations of the existing system. The system provides proper security and reduces the manual work.

### 3.4 ADVANTAGES OF PROPOSED SYSTEM

- Minimize manual data entry.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service..
- Faster and more accurate.

## **4. SYSTEM ANALYSIS**

## 4.1 INTRODUCTION TO SYSTEM ANALYSIS

System analysis is the detailed study of the various operations needed to develop a system. It is a development method that leads to the development of a new system or to modify the existing system. System analysis allows the analyst to learn about a system or process in manageable and logical way while providing a basis for ensuring that collecting details do not get over looked. The following steps are needed for analyzing a system.

**Requirement identification:** - In this step the requirements of the users are identified. It is done by discussion with the client or by interview.

**Requirement specification:** - In this the identified requirements are specified in a neat way.

**Feasibility analysis:** - It is carried out to select the best system that meets performance requirements.

**Hardware study:** - In this step the hardware requirement needed for the system to perform well is identified and specified.

**Software study:** - In this step the software requirement of the system are identified. The language used is also specified.

**System design:** - Next step is to write the code for the system. The main thing is to note is that the code should be easy to read and understand.

**System implementation and maintenance:** -Now the system is implemented and give to the client for their use. They will work with the system and specify if they need any further changes. Like any system, there is an aging process that requires periodic maintenance of hardware and software. The importance of maintenance is to continue to bring the new system to standards.



## **4.2 FUNCTIONAL SPECIFICATION**

A functional specification in systems engineering and software development in the documentation that describes the required behavior of an engineering system. The documentation typically describes what is needed by the system user as well as required properties of input and output. Thus it picks up the result of the requirement analysis stage. On more complex systems multiple levels of functional specification will typically next to each other, example. On the system level, on the module level and on the level of technical details.

## **4.3 FEASIBILITY STUDY**

Feasibility study is a system proposal according to its workability, impact on the organization, ability to meet user needs and efficient use of resources. The key considerations that are considered in feasibility analysis are:

- Economic feasibility.
- Technical feasibility.
- Behavioral feasibility.

### **4.3.1 ECONOMIC FEASIBILITY**

The proposed system is cost effective because of its compatibility, effort saving nature. The system reduces unnecessary expenses and wastage of man-hours in searching and understanding the bulky ledger reports. It benefits the user in getting hands-on experiences.

### **4.3.2 TECHNICAL FEASIBILITY**

Technical feasibility centers around the existing computer system (H/W, S/W, etc.) and to what extent it can support the proposed addition.

### **4.3.3 BEHAVIORAL FEASIBILITY**

It centers on the reach of the users. Since the system is not so complicated it is easy understandable by everyone. User training is also very easy to do. Here users don't need to have any concept of Visual Basic 6.0 for operating the system.

## **5. PROJECT OVERVIEW**

## 5.1 MODULE

There is 4 modules in our project, they are

- Login module
- Teacher Module
- Admin Module
- College Examiner module

## 5.2 MODULE DESCRIPTION

- Login  
This module helps to login to the software.
- Teacher Module  
This module helps to Input question based on modules of a subject. Teacher can also check the question repetition based on modules. If any question repeated then teacher can be replaced it. Teacher can register on software and use his user id and password to log in. If any question are same but its type are different then teacher can link that question.
- Admin  
This module helps to generate question paper from given question bank of each subject. He can also support create question paper layout and generate question paper based page layout. He can also specify questions from specific modules. He also verify the registration form of teacher based on department id.
- College Examiner  
This module helps to download generated question paper from software.

## **6. REQUIRMENT SPECIFICATION**

## 6.1 HARDWARE SPECIFICATION

One of the main steps in the design process is the selection of the software and hardware platform of the system. A Graphical User Interface in this proposed system help users to understand and manipulate complex data structures in way that would impossible in character-based system.

### Minimum Hardware Configuration Required

Processor	: Intel Celeron or above
Processor Speed	: 1 GHz
RAM Capacity	: 512 MB or above
Hard Disk Drive	: 20 GB or above
Operating System	: Windows 7 or above

## 6.2 SOFTWARE SPECIFICATION

- Front end : PHP
- Back end :MySQL5.3
- Client on pc :Windows 7 or above
- Technology used :PHP, CSS, JS, HTML

## **6.3 TECHNOLOGY SPECIFICATION**

### **ABOUT THE LANGUAGE**

Software selection is an important work in a project development cycle. Software must be selected in accordance with the application and the latest technology available. ANDROID is the best choice. A user can easily run this software. MySQL Server is used as the backend.

### **My-SQL**

My-SQL is the most popular Open Source database, is developed, distributed and supported by My-SQL AB.

#### **My-SQL is a database management system**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as My-SQL Server. Since computers are very good at handling large amounts of data, database management plays a central role in computing, as stand-alone utilities, or as parts of other applications.

#### **My-SQL is a relational database management system**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The tables are linked by defined relations making it possible to combine data from several tables on request. The SQL part of “My-SQL” stands for “Structured Query Language” the most common standardized language used to access databases.

#### **My-SQL software is Open Source**

Open source means that it is possible for anyone to use and modify. Anybody can download the My-SQL software from the Internet and use it without paying anything.

The My-SQL database server is very fast, reliable, and easy to use. It was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Though under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

## **PHP**

PHP is a general-purpose server-side scripting language originally designed for Web development to produce dynamic Web pages. It is one of the first developed server-side scripting languages to be embedded into an HTML source document, rather than calling an external file to process data. Ultimately, the code is interpreted by a Web server with a PHP processor module which generates the resulting Web page. It also has evolved to include a command-line interface capability and can be used in stand alone graphical applications. PHP can be deployed on most Web servers and also as a standalone shell on almost every operating system and platform free of charge. A competitor to Microsoft's Active Server Pages (ASP) server-side script engine and similar languages, PHP is installed on more than 20 million Web sites and 1 million Web servers.

PHP was originally created by Rasmus Lerdorf in 1995. The main implementation of PHP is now produced by The PHP Group and serves as the formal reference to the PHP language. PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.

## **HTML**

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as `<img />` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

## CSS

**CSS** stands for **Cascading Style Sheets**. CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**. CSS **saves a lot of work**. It can control the layout of multiple web pages all at once. External stylesheets are stored in **CSS files**. A cascading style sheet (CSS) is a Web page derived from multiple sources with a defined order of precedence where the definitions of any style element conflict. The Cascading Style Sheet, level 1 (CSS1) recommendation from the World Wide Web Consortium (W3C), which is implemented in the latest versions of the Netscape and Microsoft Web browsers, specifies the possible style sheets or statements that may determine how a given element is presented in a Web page.

CSS gives more control over the appearance of a Web page to the page creator than to the browser designer or the viewer. With CSS, the sources of style definition for a given document element are in this order of precedence



## JAVASCRIPT

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. While JavaScript is influenced by Java, the syntax is more similar to C and is based on ECMAScript, a scripting language developed by Sun Microsystems.

JavaScript is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a webpage has loaded without communicating with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required fields have been filled out. The JavaScript code can produce an error message before any information is actually transmitted to the server.

Like server-side scripting languages, such as PHP and ASP, JavaScript code can be inserted anywhere within the HTML of a webpage. However, only the output of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate .JS file, which may also be viewed in a browser.

## **7. SYSTEM DESIGN AND MODELING**

## 7.1 INTRODUCTION TO SYSTEM DESIGN

The creative aspect of any software is its design. The better design will be the quality and accuracy of the software that is developed. The design of an information system produces the details that state how a system will meet the requirement identified during system stage. System specialist often refers this stage as logical design. Here determines how the output is to be produced and in which format. Input data and master files have to be design to meet the requirement of the proposed output.

In any software design, initially the preliminary design is carried out, followed by a design. Relation between theoretical ideas and the resources available established during the course of design. System design is the process of design a new system. There are two steps in system design - logical and physical design. Logical design reviews the present physical system, prepares input and output specification. The physical design maps out the details of the physical system, plans the system implementation, devices a test and implementation, plan and specify any new hardware and software.

## 7.2 OBJECT ORIENTED SYSTEM

The design of object-oriented software requires the definition of multi-layered software architecture, the specification of subsystems that perform required functions and provide infrastructure support, a description of objects (classes) that form the building block of the system, and a description of the communication mechanisms that allow data to flow between layers, subsystem and objects. Object-oriented design accomplishes all the three things. The four layers of the object-oriented design pyramid are:

**The subsystem layer:** This layer contains a representation of each of the subsystems that enable the software to achieve its customer-defined requirements and to implement the technical infrastructure that support customer requirements.

**The class and the object layer:** This layer contains the class hierarchies and enables the system to be created using generalizations and increasingly more targeted specialization. This layer also contains representation of each object.

**The message layer:** This layer contains design details that enable each object to communicate with its collaborators. This layer establishes the external and internal interfaces for the system.

**The responsibility layer:** This layer contains the data structure and algorithmic design for all attributes and operations for each object.

### 7.3 SYSTEM DESIGN PROCESS

System design develops the architectural details required to build a system or product. The system design process encompasses the following activities.

- Partition the analysis model into subsystems.
- Identify the concurrency that is dictated by the problem.
- Allocate subsystem to processors and tasks.
- Develop a design for the user's interface.
- Choose a basic strategy for implementing data management.
- Identify global resources and control mechanism required to access them.
- Design an appropriate control mechanism for the system. Including task management.
- Consider how boundary conditions should be handled.
- Review and consider trade-offs.

### 7.4 OBJECT DESIGN PROCESS

During this process, we focus on the description of data structures that implement class attributes, algorithms that implement operations and messages that enable collaboration and object relationships. A variety of representations contained in the analysis model and the system design provide a specification for all operations and attributes. An algorithm is a simple computations or procedural sequence that can be implemented as a self-contained software module. Data structures are designed concurrently with algorithms. Since operations invariably manipulate the attributes of a class, the design of the data structure that best reflect the attributes will have a strong bearing on the algorithmic design of the corresponding operations.

## **7.5 MODULAR DESIGN**

This is the low-level design during which each module is designed in detail. Detailed design is concerned with specifying algorithmic detail and user interface specifications for the procedures in each module. The component level design transforms structural elements of the software architecture into procedural descriptions of software components. A design should lead to components that exhibit independent functional characteristics.

### **7.5.1 FUNDAMENTAL DESIGN**

#### **ABSTRACTION**

Abstraction permits one to concentrate on a problem at some level of generalization without regard to irrelevant low level details, use of abstraction also permits one to work with concepts and terms that are familiar in the problem environment without having to transform them to an unfamiliar structure. Two types of abstraction are there, one is procedural abstraction and data abstraction. A procedural abstraction is a named sequence of instructions that has a specific and limited function. A data abstraction is a named collection of data that describes a data object.

#### **MODULARITY**

Modularity is the single attribute of software that allows a program to be intellectually manageable. Software architecture embodies modularity, that is, software is divided into named and addressable components, called modules, which are integrated to satisfy problem requirements.

#### **PRODUCT ARCHITECTURE**

Product architecture alludes to “the overall structure of the software and the ways in which that structure provides conceptual integrity for a system”. Control hierarchy also called program structure represents the organization of control. The tree structure used to represent the control hierarchy.

## **DATA STRUCTURE**

Data structure is a representation of logical relationship among individual elements of data. Because the structure of information will invariably affect the final procedural design, data structure is very important as the program structure to the representation of the software architecture. Data structure dictates the organization, methods of access, degree of associativity, and processing alternatives for information. The organization and complexity of a data structure are limited only by the ingenuity of the designer. Scalar item array and linked list are some of the representations of the data structure.

## **SOFTWARE PROCEDURE**

Program structure defines control hierarchy without regard to the sequence of processing and decisions. Software procedure focuses on the processing details of each module individually. Procedure must provide a precise specification of processing, including sequence of events, exact, decision points, repetitive operations and even data organization structure. Information hiding suggests that modules be “characterized by design decisions that hide from all others”.

## **7.6 PHYSICAL DESIGN**

### **Data Dictionary**

The overall objective in the development of database technology has been treated data as organizational resources and as an integrated whole. Data Base Management System allows the data to be protected and organized separate from other resources. Database is defined as an integrated collection of data. Computer based systems that support centralized data management software is a software package called a Database Management System.

The most significant difference between a DBMS and other types of data management is the separation of data as seen by the program and data are stored on the direct access storage device. This is the difference between logical and physical data. An array of data items in the list is so organized that individual can be retrieved by specifying keys which are stored as part of the items or by the position of the data item in the task. A well-defined table will enable the programmer to write efficient, data independent

programs, which will not need major modifications whenever new data or storage techniques are added to the system. The design is mainly concerned with choosing 'record' types and content and set types, content and characteristics.

## 7.7 INPUT DESIGN

Input design is a design process of converting user oriented inputs to a computer based format. The keyboard is used as input media. Details are entered through data entry screens by keyboard. Outline data entry accepts commands and the data are displayed on the CRT screen for verification. The major approaches to input design are the menu and the prompt design. In each alternative, the user's options are predefined and the system is designed in a user friendly manner. Appropriate error messages are given when false details are entered. Design of a system in a menu driven fashion enables the user to select any option accordingly using simple keystrokes or mouse clicks.

The following are the features of the data entry screen for proposed system:

- 1. User friendly:** - The proposed system is designed in a user friendly manner.
  - 2. Menu driven:** - The proposed system is menu driven. This helps the user to select any option designs at any time and operations are very easy.
  - 3. Interact:** - The package is developed on the support of menu driven program. When we look through the menu items we can easily understand what is mean, so a person with little experience can also operate the system.
- 
1. Input design is the link between the information system and the users and those steps that are necessary to put transaction data into the usable form for processing data entry. The activity of putting data into the computer for processing can be activated by instructing the computer to read data from a written printed document or it can occur by keying data directly into system. The designs of input focusing on controlling the amount of input required controlling the errors, avoid delay extra steps, and keeping the process simple.
  2. The input form is highly designed with data validation, data integration and consistency with databases and application logic.

3. The users are directed with standard messages and alerts which enables them to feed the data with accuracy. It also provides short keys which make the feeling of data much simpler and easier.

Record types may contain the following data types: -

The primary key should be unique to the record. A primary key should be suitable for use as a parameter for storing values, which are available to a program. The data items, which are not, used as secondary key and which are not repeating groups.

The secondary key data items which are unique to the record and may if fixed length and a suitable code for addressing, be stored in the record rather than as a set.

## **7.8 OUTPUT DESIGN**

The output is designed in such a way that it may acquire the full satisfaction of the system user. The output devices usually used are printers and VDU. So the output can be hardcopy outputs or display outputs. As the outputs are the most important source of information to the users, better design should improve the system's relationship with us and also help in decision making. The formatted output is designed in order to understand the facts at a glance. Each column is labeled with suitable headings. Another thing that is to be considered is the output device's capability, print quality, response time, requirements etc.

## **7.9 LOGICAL DESIGN**

### **DESIGN NOTATIONS**

Design is defining a model of the new system and continues by converting this model to a new system. The method is used to convert the model of the proposed system into computer specification. Data models are converted to a database and processes and flows to user procedures and computer programs. Design proposes the new system that meets these requirements. This new system may be built by a fresh or by changing the existing system. The detailed design starts with three activities, database design, user design and program design. Database design uses conceptual data model to produce a database design. User procedure design uses those parts of the DFD outside the automation boundary to design user procedures.



## DESIGN PROCESS

Design begins when management approves the feasibility study produced during detailed analysis and authorizes the necessary funds and personnel to continue. It concludes when management approves the design and authorizes development of the actual system.

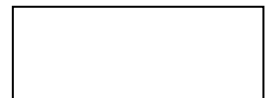
## 8.DATA FLOW DIAGRAM

The data flow diagram (DFD) was first developed by Larry Constantine as a way of expressing system requirements in graphical form, this leads to a modular design. A DFD, is also known as a single 'Bubble Chart', has the purpose of clarifying system requirements and identify major transformation that will become programs in system design. A DFD consist of a series of bubbles joined by lines. The bubbles represent the data transformations and the line represents data flow in the system.

### 8.1 DFD SYMBOLS

The DFDs are constructed from four basic symbols. They are,

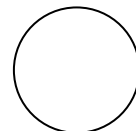
- A SQUARE defines a source or destination of system data.



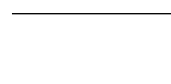
- An ARROW identifies the flow of data.



- A CIRCLE or a BUBBLE represents a process that transforms incoming data-flow into outgoing flows.



- An OPEN-ENDED RECTANGLE is a data source- a temporary repository of data.

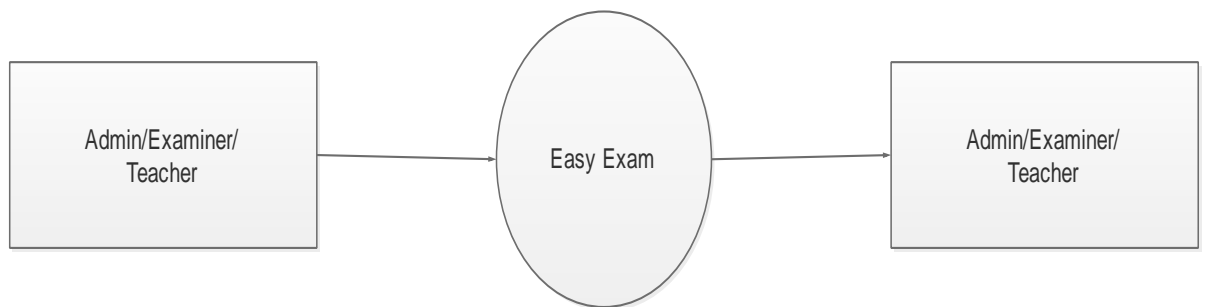


## 8.2 CONSTRUCTING A DFD

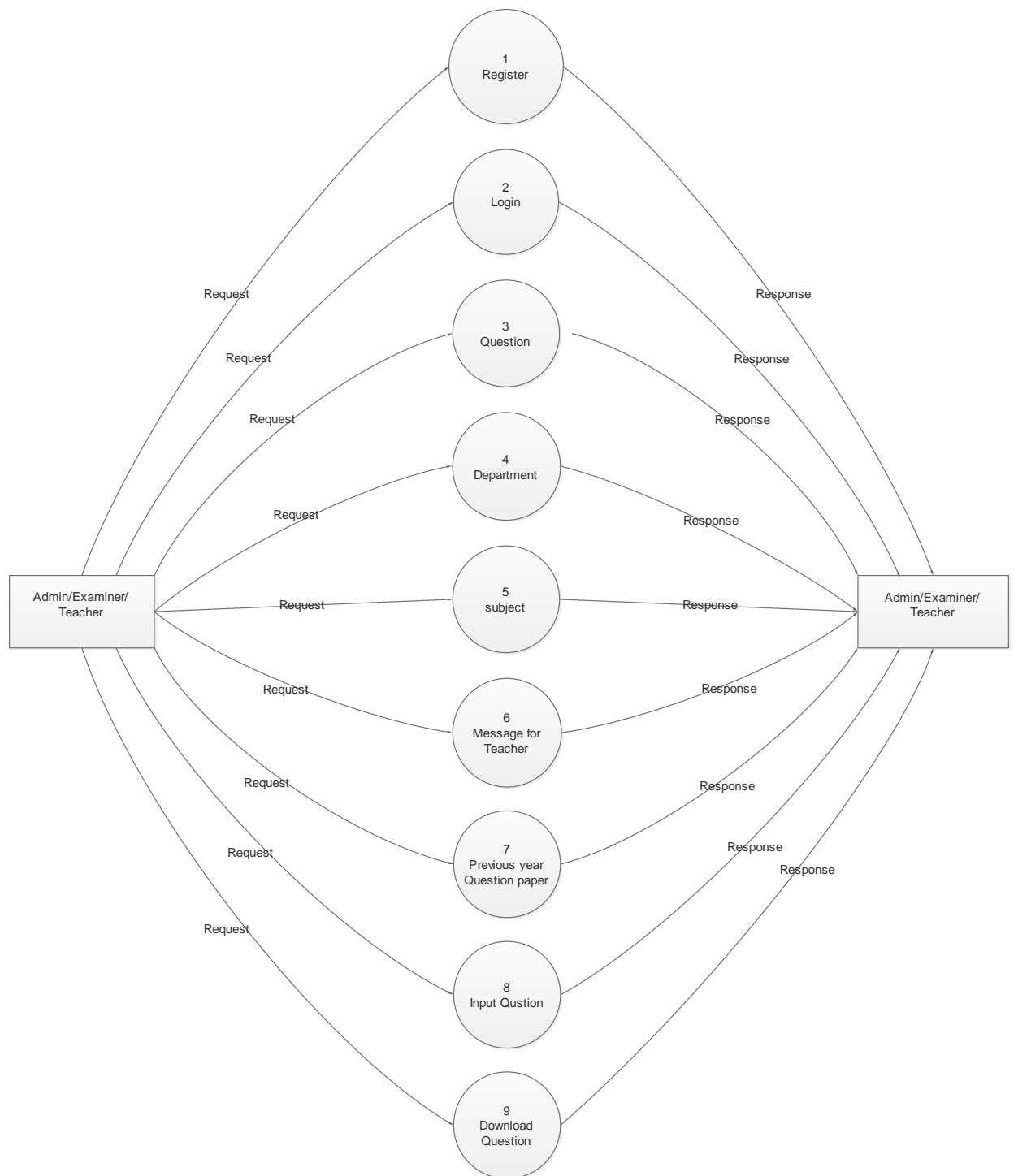
Several rules of thumb are used in drawing DFDs:

1. Process should be named and numbered. Each should be representative of the process.
2. The direction of flow is from top to bottom and from left to right.
3. When a process is exploded in to lower level details, they are numbered.
4. The name of data stores, sources and destinations are written in capital letters.  
Process and data flow names have the first letter of each word capitalized.

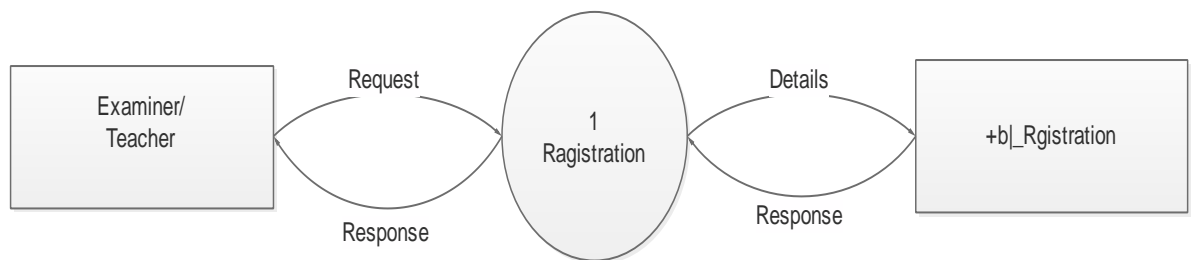
## ZERO LEVEL



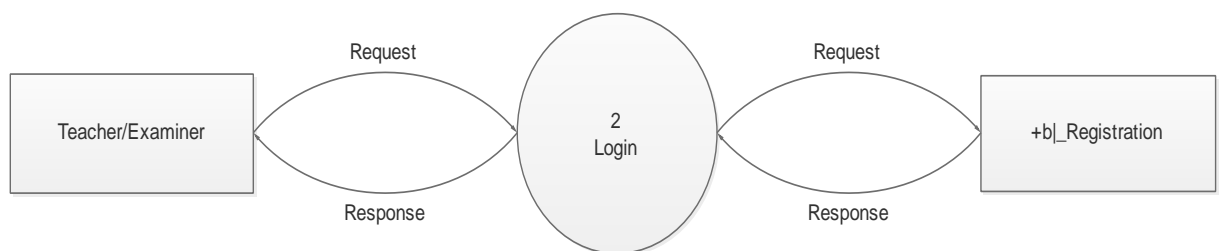
## FRIST LEVEL

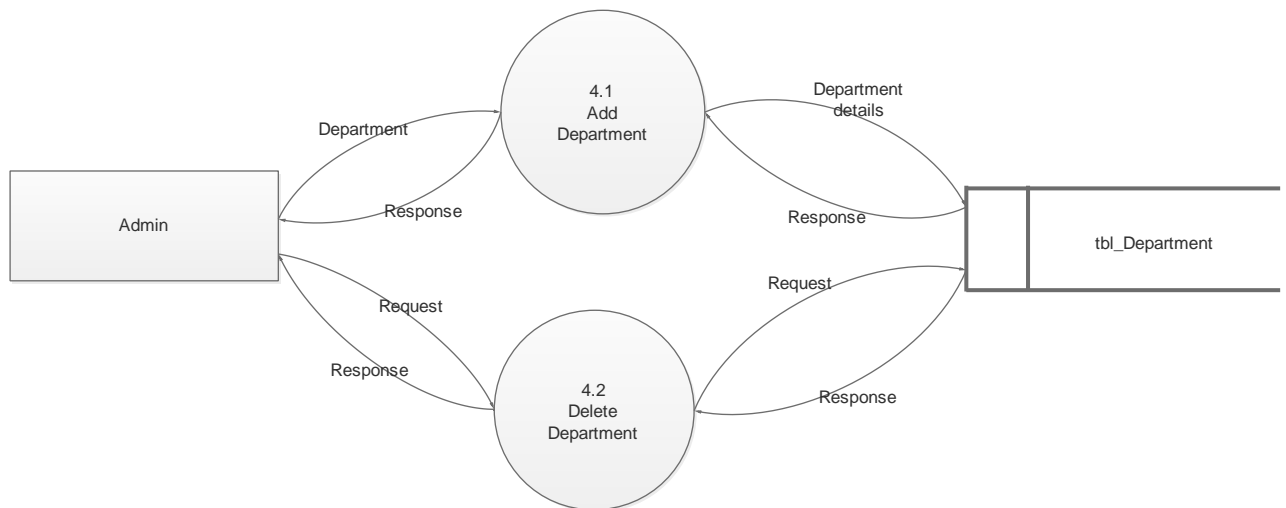
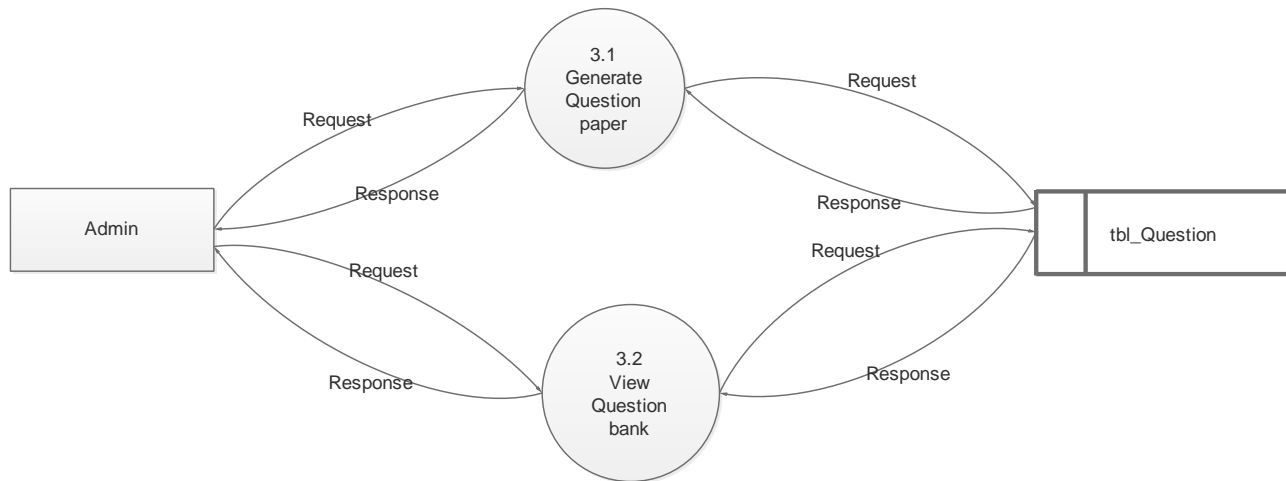


## Registration

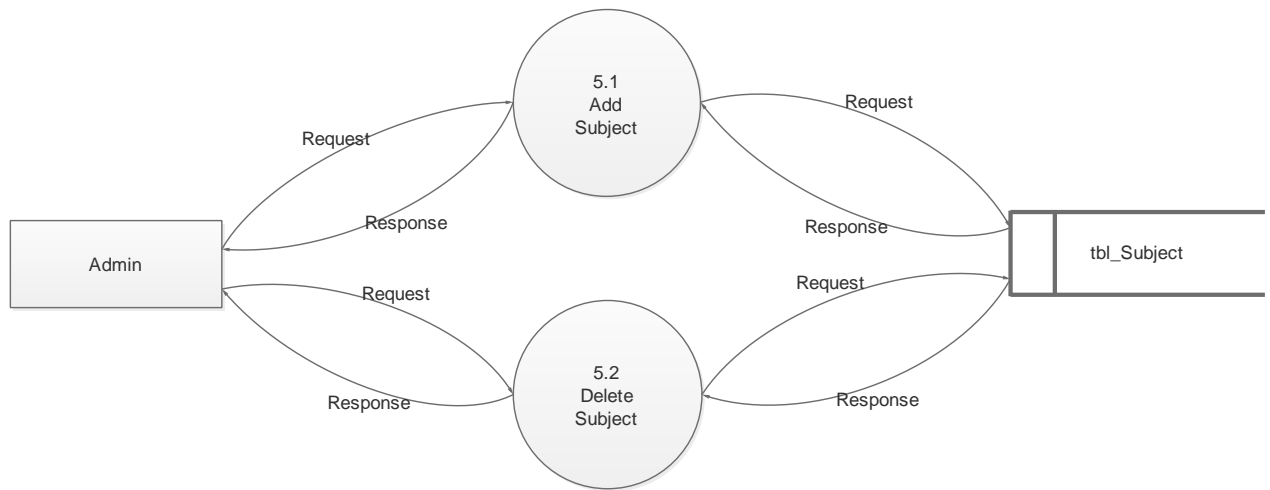


## Login

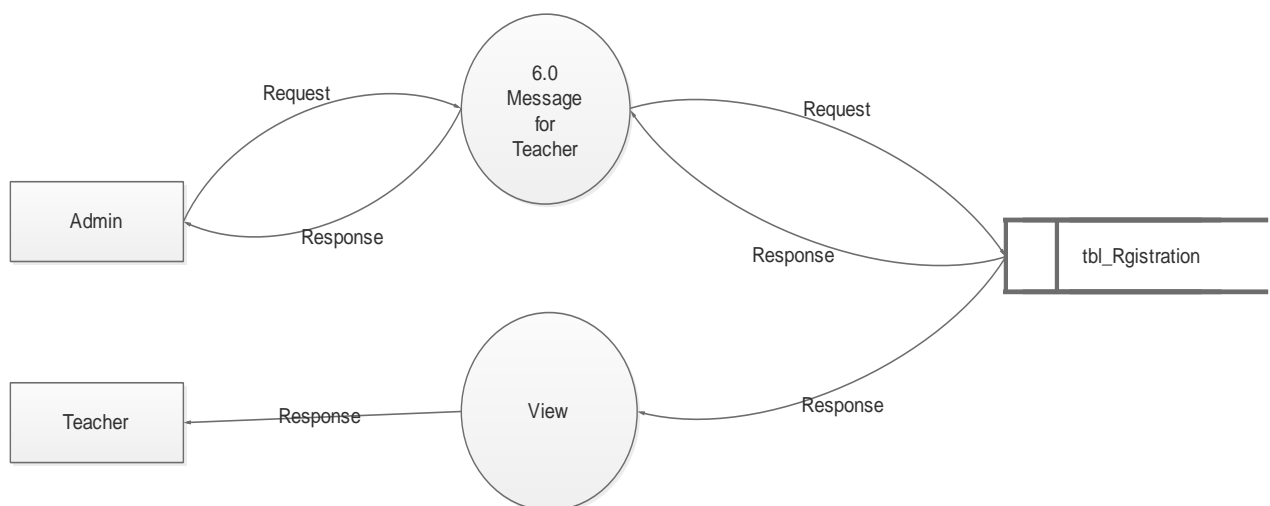




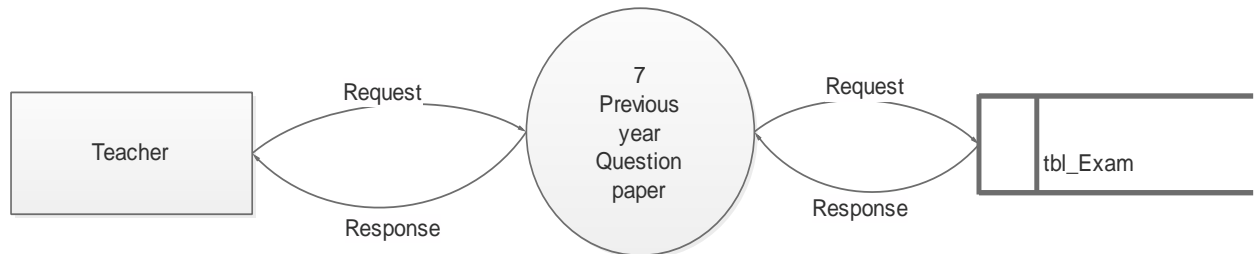
## Subject for Admin



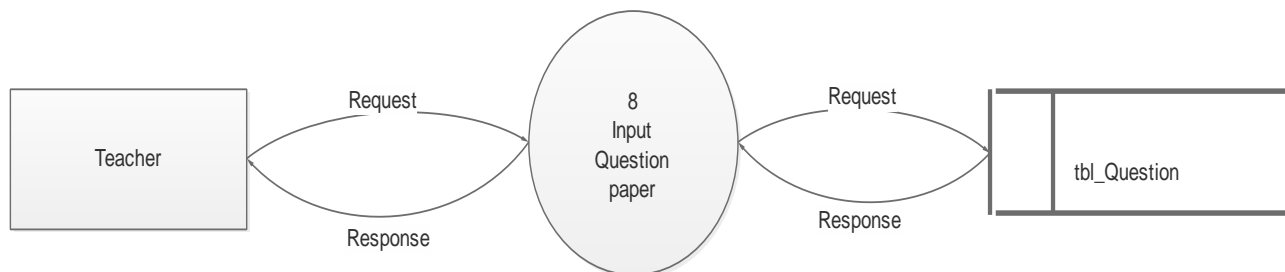
## Message for Teacher



## Previous year Question paper for Teacher

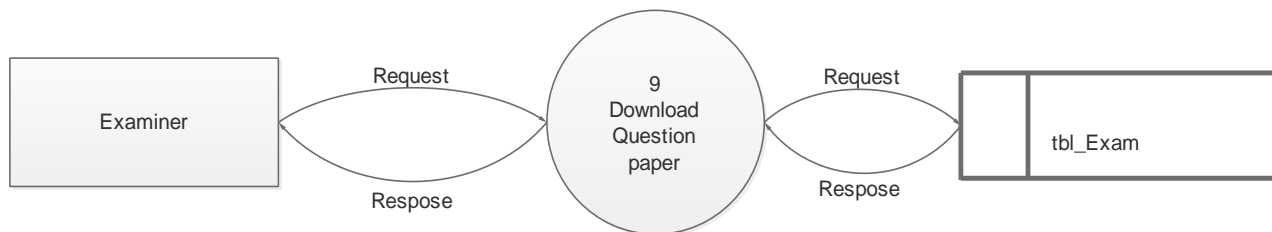


## Input Question for Teacher





## Download Question paper for Examiner

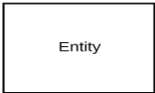


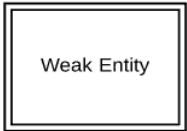
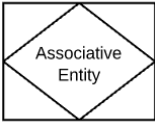
## 8. ENTITY RELATIONSHIP DIAGRAM

Entity-relationship diagrams (ERD) are essential to modeling anything from simple to complex databases, but the shapes and notations used can be very confusing. This guide will help you to become an expert in ER diagram notation, and you will be well on your way to model your own database.

### ERD entity symbols

Entities are objects or concepts that represent important data. Entities are typically nouns such as product, customer, location, or promotion. There are three types of entities commonly used in entity relationship diagrams.



Entity Symbol	Name	Description
	Strong entity	These shapes are independent from other entities, and are often called parent entities, since they will often have weak entities that depend on them. They will also have a primary key, distinguishing each occurrence of the entity.

Entity Symbol	Name	Description
	Weak entity	Weak entities depend on some other entity type. They don't have primary keys, and have no meaning in the diagram without their parent entity.
	Associative entity	Associative entities relate the instances of several entity types. They also contain attributes specific to the relationship between those entity instances.

### ERD relationship symbols


Within entity-relationship diagrams, relationships are used to document the interaction between two entities. Relationships are usually verbs such as assign, associate, or track and provide useful information that could not be discerned with just the entity types.




Relationship Symbol	Name	Description
<hr/>		

Relationship Symbol	Name	Description
	Relationship	Relationships are associations between or among entities.
	Weak relationship	Weak Relationships are connections between a weak entity and its owner.

### ERD attribute symbols

ERD attributes are characteristics of the entity that help users to better understand the database. Attributes are included to include details of the various entities that are highlighted in a conceptual ER diagram.

Attribute Symbol	Name	Description
	Attribute	Attributes are characteristics of an entity, a many-to-many relationship, or a one-to-one relationship.

Attribute Symbol	Name	Description
	Multivalued attribute	Multivalued attributes are those that are can take on more than one value.
	Derived attribute	Derived attributes are attributes whose value can be calculated from related attribute values.
	Relationship	Relationships are associations between or among entities.

## **9. ENTITY RELATIONSHIP DAIAGRAM**



## 10. Table Design

### Department Table

No.	Name	Data Type	Constrains	Description
1	dept_id	int	Primary Key	Department id
2	dept_name	varchar(50)	Not null	Department Name

### Login Table

No.	Name	Data Type	Constrains	Description
1	log_id	int(11)	Pimary Key	Login id
2	username	varchar(50)	Not null	username
3	password	varchar(10)	Not Null	password
4	role	varchar(25)	Not null	Role of user
5	status	int(11)	Not null	Status of user



## Log\_time Table

No.	Name	Data Type	Constraints	Description
1	loguser_id	int(11)	Foreign key	Examiner login id
2	time	timestamp	Not Null	on update CURRENT_TIMESTAMP

## PDF Files Table

NO.	Name	Data Type	Constraints	Description
1	u_id	int(11)	Primary Key	Id of file
2	examination_name	varchar(100)	Not Null	Name of Examination
3	subject_name	int(50)	Foreign key	Id of department
4	course_name	int(50)	Foreign key	Id of Subject
5	examination_date	date	Not Null	Date of Examination
6	File	varchar(50)	Not Null	PDF file

## Question Table

No.	Name	Data Type	Constraints	Description
1	q_id	int(11)	Primary Key	Id of question
2	sub_id	int(11)	Foreign key	Id of Subject
3	Question	varchar(100)	Not Null	Question
4	question_complex	varchar(10)	Not Null	Complexity of Question
5	question_type	int(11)	Not Null	Type of Question
6	Module	int(11)	Not Null	Module of subject
7	teacher_id	int(11)	Foreign key	Id of teacher
8	dept	int(50)	Foreign key	Id of department

## Registration for Examiner Table

No.	Name	Data Type	Constraints	Description
1	examiner_id	int(11)	Primary Key	Id of Examiner
2	name	varchar(25)	Not Null	Name of Examiner
3	college_name	varchar(75)	Not Null	College name
4	phno	varchar(10)	Not Null	Phone No. of Examiner
5	email	varchar(75)	Not Null	Email of Examiner
6	password	varchar(12)	Not Null	Password of Examiner

## Registration for Teacher Table

No.	Name	Data Type	Constraints	Description
1	reg_id	int(11)	Primary Key	Id of Teacher
2	fname	varchar(25)	Not Null	Name of Teacher
3	cname	varchar(75)	Not Null	College name
4	dept	int(75)	Foreign key	Id of department
5	phno	varchar(10)	Not Null	Phone Number
6	email	varchar(50)	Not Null	Email Address
7	password	varchar(15)	Not Null	Password

## Subject Table

No.	Name	Data Type	Constraints	Description
1	sub_id	int(11)	Primary Key	Id of Subject
2	dept_id	int(11)	Foreign key	Id of Department
3	subject_name	varchar(35)	Not Null	Name of Subject
4	course_code	varchar(12)	Not Null	Course Code of Subject
5	semester	int(11)	Not Null	Semester

## 11. CODING

### Code for DB Function

```
<?php

include_once('controllers/connection.php');

$obj=new Connection;

if(isset($_POST['login']))

{

    echo $username=$_POST["username"];

    echo $password=$_POST["password"];

    $table="log_admin";

    $type="admin";

    $data=$obj->check_login($username,$password,$table,$type);

    if($data==0)

    {

        header('location:home.php');

    }

    elseif($data==1)

    {

        @session_start();

        $status=$_SESSION['status'];

        if($status=='1')

        {

            header('location:error500.php');

        }

        else{
```

```
        header('location:home.php');

    }

}

elseif($data==2)

{

    @session_start();

    $time=date('Y-m-d h:i:sa');

    //$time=date('d-m-Y h:i:sa');

    $id=$_SESSION['reg_id'];


    $obj->execute("INSERT INTO `log_time`(`loguser_id`, `time`) VALUES
($id,$time)");


    header('location:home.php');

}

else{

    header('location:login.php?error="Invalid Authentication"');

}

}

else if(isset($_POST['t_reg']))

{
```

```
$log=array(

    //log_id'=>$_POST[''],

    'username'=>$_POST['email'],

    'password'=>$_POST['password'],

    'role'=>'teacher',

    'status'=>'1'

);

$id=$obj->insert2('log_admin',$log);


$regi=array(

    'reg_id'=>$id,

    'fname'=>$_POST['fname'],

    'cname'=>$_POST['cname'],

    'dept'=>$_POST['dept'],

    'phno'=>$_POST['phno'],

    'email'=>$_POST['email'],

    'password'=>$_POST['password'],

    're_password'=>$_POST['re_password']

);

if($obj->insert('reg_teacher',$regi))

{

    header('location:reg.php');
```

```
        }

        else{

            header('location:home.php');

        }

    }

    else if(isset($_POST['t_reg1']))

    {

        $log=array(

            // 'log_id'=>$_POST[''],

            'username'=>$_POST['email'],

            'password'=>$_POST['password'],

            'role'=>'teacher',

            'status'=>'1'

        );

        $id=$obj->insert2('log_admin',$log);

        $regi=array(

            'reg_id'=>$id,

            'fname'=>$_POST['fname'],

            'cname'=>$_POST['cname'],

            'dept'=>$_POST['dept'],

            'phno'=>$_POST['phno'],

            'email'=>$_POST['email'],
```



```
        'password'=>$_POST['password'],
        're_password'=>$_POST['re_password']
    );

    if($obj->insert('reg_teacher',$regi))
    {
        header('location:signup.php');
    }
    else{
        header('location:home.php');
    }
}
else if(isset($_POST['add_ques']))
{
    @session_start();

    $id=$_SESSION['id'];

    $dept=$_SESSION['dept'];

    $regi=array(
        'sub_id'=>$_POST['sub_id'],
        'question'=>$_POST['question'],
        'question_complex'=>$_POST['question_complex'],
        'question_type'=>$_POST['question_type'],
        'module'=>$_POST['module'],
        'teacher_id'=>$id,
        'dept'=>$dept
    );
}
```

```
        if($obj->insert('question',$regi))
        {
            header('location:add_question.php');
        }
        else{
            header('location:home.php');
        }
    }
else if(isset($_GET['d_teacher']))
{
    $table='reg_teacher';
    $condition='reg_id='.$_GET['d_teacher'];
    $table1='log_admin';
    $condition1='log_id='.$_GET['d_teacher'];
    $result=$obj->deletion($table1,$condition1);
    $result=$obj->deletion($table,$condition);
    if($result=='ok')
    {
        header('location:remove-teacher.php');
    }else{
        header('location:home.php');
    }
}
```

```
}

else if(isset($_GET['d_examiner']))

{

    $table='reg_examiner';

    $table1='log_admin';

    $condition='examiner_id='.$_GET['d_examiner'];

    $condition1='log_id='.$_GET['d_examiner'];

    $result=$obj->deletion($table,$condition);

    $result=$obj->deletion($table1,$condition1);

    if($result=='ok')

    {

        header('location:remove_examiner.php');

    }else{

        header('location:home.php');

    }

    //?error//?success=OK

}

else if(isset($_POST['e_reg']))

{

    $log=array(

        //log_id=>$_POST[''],

        'username'=>$_POST['email'],

        'password'=>$_POST['password'],
```

```
        'role'=>'examiner','status'=>'0'

    );

    $id=$obj->insert2('log_admin',$log);

    $regi=array(

        'examiner_id'=>$id,

        'name'=>$_POST['name'],

        'college_name'=>$_POST['college_name'],

        'phno'=>$_POST['phno'],

        'email'=>$_POST['email'],

        'password'=>$_POST['password']

        //'re_password'=>$_POST['re_password']

    );

    if($obj->insert('reg_examiner',$regi))

    {

        //alert("Successfull");

        header('location:reg_examiner.php');

    }

    else

    {

        header('location:home.php');

    }

}

else if(isset($_POST['add_dep']))
```

```
{

    $dep=array(

        'dept_id'=>$_POST['dept_id'],

        'dept_name'=>$_POST['dept_name']

    );

    if($obj->insert('department',$dep))

    {

        header('location:add_dept.php');

    }

    else{

        header('location:home.php');

    }

}

else if(isset($_POST['add_sub']))

{

    $regi=array(

        'dept_id'=>$_POST['dept_id'],

        'subject_name'=>$_POST['subject_name'],

        'course_code'=>$_POST['course_code'],

        'semester'=>$_POST['semester']

    );

    if($obj->insert('subject',$regi))

    {

        header('location:add_subject.php');
```

```
        }

        else{

                header('location:home.php');

        }

}

else if(isset($_POST['send']))

{

        $message=$_POST['message'];

        $member=$_POST['member'];

        $page='message1.php?success';

        if($member=='examiner')

        {

                $result=$obj->getAll('SELECT phno FROM reg_examiner');

                foreach($result as $number)

                {

                        extract ($number);

                                $obj->sms($phno,$message,$page);

                }

        }

        else if($member=='teacher')

        {

                $result=$obj->getAll('SELECT phno FROM reg_examiner');
```

```
        foreach($result as $number)
        {
            extract ($number);

            $obj->sms($phno,$message,$page);

        }
    }
else
{

}

}

else if(isset($_GET['u_status']))
{

    $field=array('status'=>'0');

    $table='log_admin';

    $condition="log_id='".$_GET['u_status']."'";

    $others=array('page'=>'teacher_verification.php');

    $obj->update($field,$table,$condition,$others);

}

else if(isset($_GET['d_history']))
{
```

```
$table='log_time';

$condition='loguser_id='.$_GET['d_history'];

$result=$obj->deletion($table,$condition);

if($result=='ok')

{

    header('location:login_history.php');

}else{

    header('location:home.php');

}

}

else if(isset($_POST['q_upload']))

{

    $dep=array(

        'dept_id'=>$_POST['dept_id'],

        'dept_name'=>$_POST['dept_name']

    );

    if($obj->insert('department',$dep))

    {

        header('location:add_dept.php');

    }

    else{
```



```
        header('location:home.php');

    }

}

else if(isset($_POST["qb_upload"])){

    array_pop($_POST); //omits the button when button name comes last or
    else use "unset"

    extract($_POST);

    $data=$_POST;

    $pathname=image_upload($_FILES['image']['name']);

    $data['file']=$pathname;

    $result=$obj->insert_no_redirect("pdf_files",$data);

    if($result=='OK'){

        header('location:upload_question_bank.php');

        exit;

    }

    else{

        header('location:upload_question_bank.php');

        exit;

    }

}

function image_upload($imgTp) //: Anonymous function

{

    $fileinfo = pathinfo($imgTp);

    //getting the file extension
```

```
$extension = $fileinfo['extension'];

if (($extension != "jpg") && ($extension != "pdf") &&
($extension != "jpeg") && ($extension != "png") && ($extension != "gif"))

{

    echo 'Unknown image format.';

}

//jpg-jpeg

if($extension=="jpg" || $extension=="jpeg" || $extension=="pdf" )

{

    $uploadedfile = $_FILES['image']['tmp_name'];

    $temp = explode(".", $_FILES["image"]["name"]);

    $newfilename = round(microtime(true)) . '.' .

end($temp);

    $filename1 = "uploads/". $newfilename;

    move_uploaded_file($uploadedfile, "$filename1");

    echo $filename1;

    return $filename1;

}

}
```

```
if (! empty($_POST["dept_id"])) {

    $query = "SELECT * FROM department WHERE dept_id = " . $_POST["dept_id"] .

    " order by name asc";
```

```
$results = $obj->getAll($query);

?>

<option value disabled selected>Select Department</option>

<?php

    foreach ($results as $dept) {

        ?>

        <option value="<?php echo $dept["dept_id"]; ?>"><?php echo $dept["dept_name"];
        ?></option>

    <?php

        }

    }

    if (! empty($_POST["sub_id"])) {

        $query = "SELECT * FROM subject WHERE dept_id = '" . $_POST["sub_id"] . "'";

        $results = $obj->getAll($query);

        ?>

        <option value disabled selected>Select Subject</option>

        <?php

            foreach ($results as $sub) {

                ?>

                <option value="<?php echo $sub["sub_id"]; ?>"><?php echo $sub["subject_name"];
                ?></option>

            <?php

                }

            }

        }
```

## **12. SYSTEM TESTING & IMPLEMENTATION**

## **12.1 SYSTEM TESTING**

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all system elements have been properly integrated and perform allocated functions.

Testing is the final verification and validation activity within the organization itself. Testing is done to achieve the following goals; to affirm the quality of the product, to find and eliminate any residual errors from previous stages, to validate the software as a solution to the original problem, to demonstrate the presence of all specified functionality in the product, to estimate the operational reliability of the system. During testing the major activities are concentrated on the examination and modification of the source code

## **12.2 TESTING METHODS**

### **12.2.1 UNIT TESTING**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

### **12.2.2 INTEGRATION TESTING**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

### **12.2.3 USER ACCEPTANCE TESTING**

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly in touch with the prospective system users at time of developing and making changes wherever required is done in regard to the following point:

Input Screen design

Output Screen design

Menu driven system

### **12.2.4 OUTPUT TESTING**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. The outputs generated or displayed by the system under consideration are tested by asking the users about the format required by them. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

## **12.2.5 VALIDATION TESTING**

Validation checks are performed on the following fields.

Text Field: The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field: The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

## **12.2.6 IMPLEMENTATION PROCEDURE**

The Implementation procedures used for this system contains following steps.

- ☐ Test Plans
- ☐ training
- ☐ Hosting

### **12.2.6.1 TEST PLANS**

The implementation of a computer based system requires that test data be prepared and that the system and its elements be tested in planned, structured

manner. The computer program components are a major subsystem of the computer-based information system, and particular attention should be given to the testing of this system element as it is developed.

#### **12.2.6.2 TRAINING**

Training have to be given for the administrator and it is only the matter of getting familiar with two web forms and the settings (what each of the setting mean). For the users we cannot train the personally. The system is developed in such a way that anybody who is familiar with using internet based application such as email can use easily. Furthermore, they can use the FAQ help also to make their job easier.

#### **12.2.6.3 HOSTING**

This system has one great feature which is its hassle free installation. With many other web based application we have to do a lot of work to get the application up and running. Here it is so easy that the system will be already to use with just a click of the mouse. Tables will be created, Xml files will get created and the necessary file hierarchies will also be created. But if the customer is not using a third party hosting service and they are hosting it on their own server they have to make sure that the prerequisites are already installed on the server.

#### **12.2.7 TESTING OBJECTIVES**

Test objectives are to make the software free from errors, to make it reliable for the future use.

1. To find uncovered errors based on requirements



2. Ensure the product is bug free before shipment.
3. 'Quality is ensured'; Objective of testing is to find out as many as error as possible and to make software more reliable and easily maintainable. A good test is one which uncovers an error. Testing leads to a software on which we can say with ensured that this software is reliable and is tested and will work % age of confidence depends upon the amount of testing. One thing to be noted is that we cannot test exhaustively because the amount of test cases is infinity so we choose optimal amount of the testing not too much or too less and depending upon closeness of the testing to the optimal level we decides the age of confidentially.

## 12.3 SYSTEM IMPLEMENTATION

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled, it can cause chaos and confusion.

Implementation includes all those activities that take place to convert from the old system to the new one. The new system may be totally new, replacing an existing manual or automated system or it may be a major modification to an existing system. Proper implementation is essential to provide a reliable system to meet the organization requirements. Successful implementation may not guarantee improvement in the organization using the new system, but improper installation will prevent it. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert

from the old system to the new system. The system can be implemented only after thorough testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system.

The most crucial stage is achieving a new successful system and giving confidence on the new system for the user that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover. The more complex the system being implemented, the more involved will be the system analysis and the design effort required just for implementation. The system implementation has three main aspects. They are education and training, system testing and changeover.

- ☐ Careful planning.
- ☐ Investigation of system and constraints.
- ☐ Design of methods to achieve the changeover.
- ☐ Training of the staff in the changeover phase

#### **□□.4 POST IMPLEMENTATION REVIEW**

We have done this project with soul intention to manage an Automatic Question paper generator. The main advantage of this project save time and project manages Question paper which is done Online .

The purpose of this post implementation review is to provide an overview of the project to implement an Automatic Question paper generator. The report also identifies the future recommendations to ensure an effective and efficient network is maintained to meet the requirements of this project.

I have successfully achieved my goal in developing a question paper generator. This system is efficient and effective in handling the college. The working of this system had done successfully and in effective manner.

## **13. SYSTEM MAINTANCE**

## 13.1SYSTEM MAINTANCE

Maintenance is an activity, which is made for environmental changes, which may affect either the computer or other parts of computer-based system. It includes both the improvement of the system functions and the correction of faults, which arise during the operation of a system. The maintenance includes a wide range of activities including correction of coding and design errors, updating the documentation and test data and upgrading the user support. Maintenance is always necessary to keep the software usable. Periodic maintenance is required for the hardware also. For the periodic maintenance of the software and hardware, there is an aging process.

Mainly there are four types of software maintenance activities. They are:

- ☐ Corrective maintenance
- ☐ Adaptive maintenance
- ☐ Perfective maintenance
- ☐ Preventive maintenance

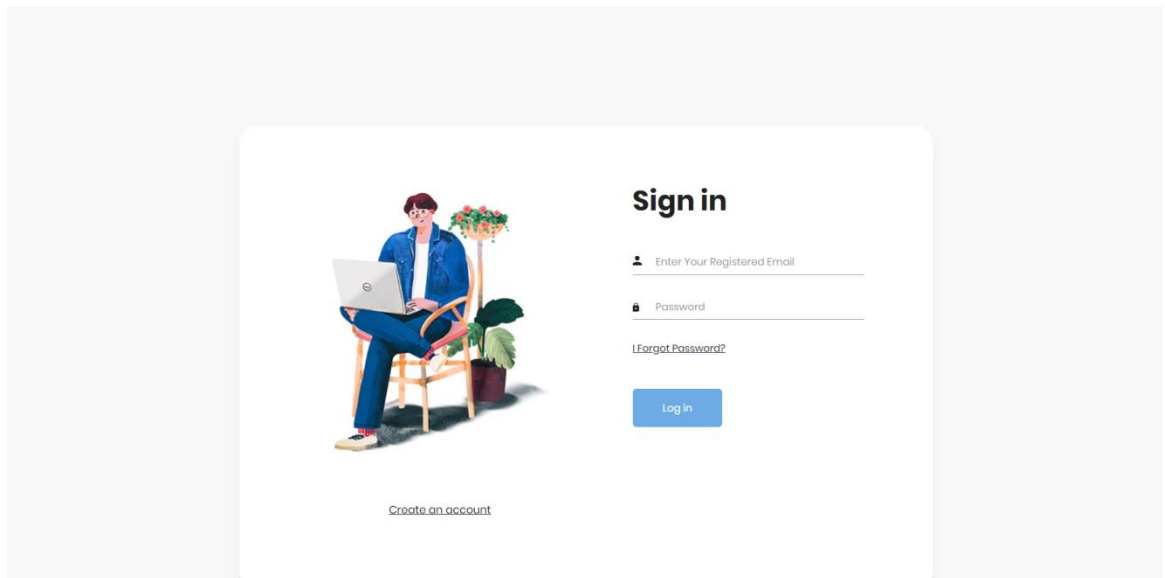
The Corrective maintenance uncovers all latent errors in a large software system. It is a process that includes the diagnosis and correction of one or more errors. Adaptive maintenance is the activity that modifies software to properly interface with a changing environment. Adaptive maintenance is needed because of the rapid change that is encountered in every aspect of computing, such as changes in the user requirement, changes in the platform or changes in external interface. Perfective maintenance is performed to satisfy requests like recommendations for new capabilities and modification to existing function. This occurs when a software package is successful. Preventive maintenance occurs when software is changed to

improve future maintainability or reliability or to provide a better basis for future enhancements. The need for maintenance has not come forward as the current project is yet to be released. So the need for reviews is also has less importance.

Software review is a 'filter' for the software engineering process. The review is applied at various points during software development and serves to uncover defects that can then be removed. Software review serves to purify the software engineering activities such as analysis, design and coding.

## **14. SCREEN SHOTS**

## 1.Screen shot for login



## Screen shot for Enumerator registration



## Sign up

Your Name

College Name

Name of Department

Mobile No.

Your Email


Password

Repeat your password

☐

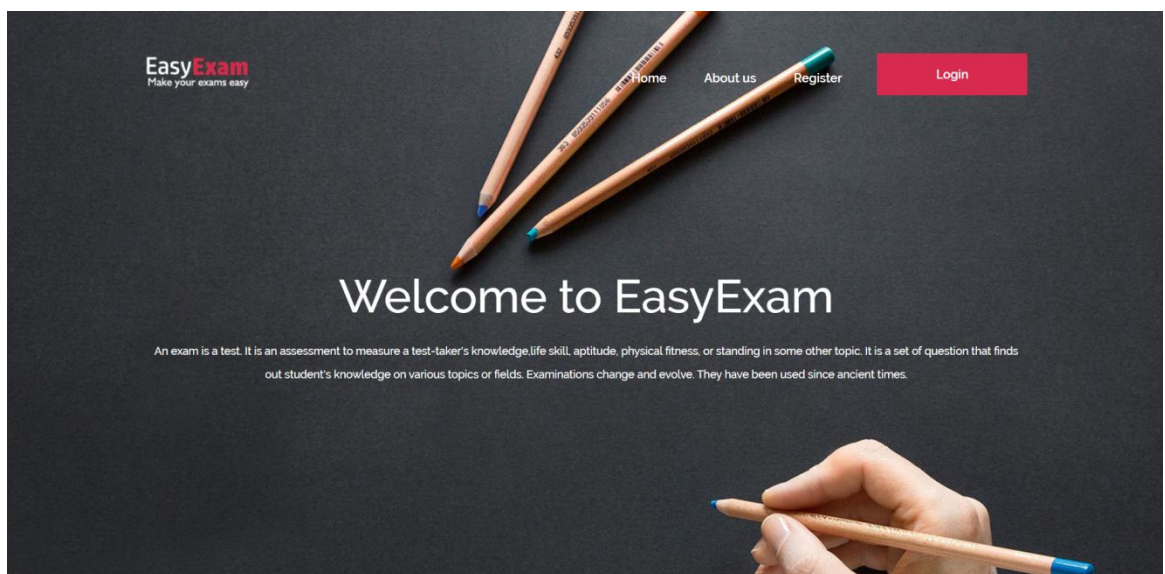
I agree all statements in [Terms of service](#)

Register

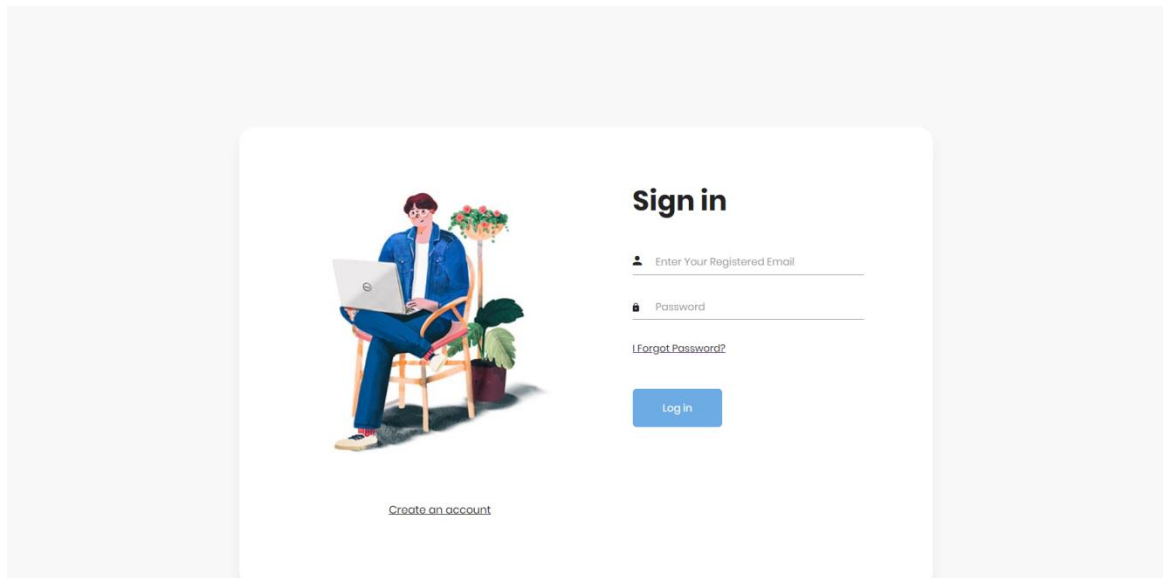


I am already member

## Screen shot for Home page



## Screen shot for Admin Login



## Screen shot for Question paper generator

EasyExam

vishnu@gmail.com  
admin

Search...

MAIN NAVIGATION

- Examiner Settings
- Teachers Settings
- Department Settings
- Generate Question Paper
- View Question Bank
- Message for Teachers
- Upload Question Bank

Teacher Main Navigation

- Input Question Bank
- Previous Question Papers
- Profile Settings

Examiner Main Navigation

- Download Question Papers

Generate Question Paper for Examination

Home > Forms > Advanced Elements

Input Question Paper Details

College Name  
Name of College

Name of Examination  
Name of Examination with Month and Year

Course Name  
Department of Commerce

Semester  
1

Course Code  
Course Code

Name of Subject  
Accounting

Number of Questions in Part A  
--

Number of Questions in Part B  
--

Number of Questions in Part C  
--

Time Duration of Examination

Total Mark of Examination

Question Paper Code

Module for Examination

☐ Module No.1 ☐ Module No.2 ☐ Module No.3 ☐ Module No.4 ☐ Module No.5

Cancel

GO

## Screen shot for Add question

**EasyExam**

vishnu@gmail.com admin

Search...

MAIN NAVIGATION

- Examiner Settings
- Teachers Settings
- Department Settings
- Generate Question Paper
- View Question Bank
- Message for Teachers
- Upload Question Bank
- Input Question Bank

**Add Questions**

Input Question to Question Bank

Subject Name:

Select Module:

Enter Question:

Question Complexity:

Question Type:

Cancel GO

## Screen shot for question details

**Question Details**

IT and Environment

Show 10 entries

Search:

Question	Question Complexity	Question Type	Module	Delete
" Education technology can improve the traditional way of teaching, but cannot replace the human tou	hard	2 Marks	3 Module	Delete
Define Cyber Crime.	easy	2 Marks	3 Module	Delete
Define green computing	easy	2 Marks	4 Module	Delete
Define the term information age.	easy	2 Marks	3 Module	Delete
Describe about the principal organs of UN.	easy	5 Marks	5 Module	Delete
Expand CERD	easy	2 Marks	5 Module	Delete
Expand ICCPR.	easy	2 Marks	5 Module	Delete
Expand ICESCR.	easy	2 Marks	5 Module	Delete
Expand UDHR.	easy	2 Marks	5 Module	Delete
Explain about common types of cyber threats	easy	5 Marks	3 Module	Delete

Showing 1 to 10 of 61 entries

Previous 1 2 3 4 5 6 7 Next

## Screen Shot for Upload question

---

## **15. CONCLUSION**

## **15.1 CONCLUSION**

This application is developed for College. It provides many features to its Teacher's and college. The difficulty of developing question paper has been completely eliminated by reserving via the proposed system.

Our project successfully reduces the difficulty of developing question paper generator and simplify the process times. Our project helps in easily developing Question paper generator. Our project saves the time of Teacher's and Examiner.

## **16. BIBLIOGRAPHY**

- “ System Analysis and Design”-**E Award:** *Golgothia, 3<sup>rd</sup> Edition*
- “Elements of System Analysis”-**Marvin Gore:***Golgothia, 4<sup>rd</sup> Edition*
- “SQL Server 7 - A Beginners guide”-**DusanPetcovic:** *Osborne, 1999*
- “*Android programming for beginners*”-**John Horton**
- “*Java programming*”-**Horstmann,** *core java 2<sup>nd</sup> Edition*

## 17. WEBLIOGRAPHY

- ☐ [www.wikipedia.com](http://www.wikipedia.com)
- ☐ [www.w3schools.com](http://www.w3schools.com)
- ☐ [stackoverflow.com](http://stackoverflow.com)

