ONLINE EXAMINATION SYSTEM

TEAM:

Reddi Akhil (317506402001)

Behara Akhila (317506402002)

Addanki Asish (317506402003)

Nirujogi Chaitanya Prasad (317506402004)

Tangudu Hanishma(317506402011)

DEPARTMENT OF COMPUTER SCIENCE AND SYSTEMS ENGINEERING
ANDHRA UNIVERSITY COLLEGE OF ENGINEERING (A)
ANDHRA UNIVERSITY

CERTIFICATE

This is to certify that this bonafide project work entitled
"ONLINE EXAMINATION SYSTEM" done by

Reddi Akhil (317506402001) ,Behara Akhila (317506402002),

Addanki Asish (317506402003) , Nirujogi Chaitanya Prasad

(317506402004),Tangudu Hanishma(317506402011) ,

Students of 3/6 CSSE, Department of computer science & systems engineering, Andhra University college of engineering (A).

Prof. Kuda Nageswara Rao
Head of the Department
Dept of CS&SE
AUCE(A)
Visakhapatnam

1. INTRODUCTION

1.1 Introduction to Project

1.2 Objectives of Project

2. SYSTEM ANALYSIS

- 2.1 Existing System
- 2.2 Proposed System
- 2.3 Feasibility Study
- 2.3.1 Technical Feasibility
- 2.3.2 Operational Feasibility
- 2.3.3 Economic Feasibility

3. SOFTWARE REQUIREMENT SPECIFICATION

- 3.1 Functional Requirements
- 3.2 Non Functional Requirements

4. ARCHITECTURE AND DESIGN OF PROPOSED SYSTEM

- 4.1 System architecture
- 4.2 Flow Chart
- 4.3Database Design
- 4.4 Data Flow Diagrams
- 4.5 UML Diagrams

5. IMPLEMENTATION (coding)

- 5.1 Introduction to technology
- 5.2 Sample code

6. TESTING

- 6.1 Testing Strategy

INTRODUCTION

The whole process of assigning test and evaluating their scores after the test, was done manually till date. It is very time consuming . Also it is difficult

to keep the answer sheets being generated as well as the maintenance of the record of each examination. The chance of loss of records is high and record searching is difficult. Result processing takes more time and the presence of more invigilators is a must if there are more number of students to write the exam.

Objectives of the project:

The objective of the project is to conduct an online examination and allot the marks there itself, and allow the user to view the number of correct answers he chose out of total number of questions on the display.

SYSTEM ANALYSIS

Existing system

The Existing system of conducting examination process is manual. It has so many problems. So we introduce a new system, which is fully computerized. Existing system is a large man power process and is difficult to implement. Working of existing system is given below: -

Student Registration is the first process . As the part of the registration, the student has to enter his name, address etc into the registration form. After the registration, make the question papers and it will give to the prospective student. The question papers contain total mark, subject, duration, questionpaper etc. A group of person does evaluation of answer sheet. After the evaluation of the Answer sheet , the result is published. And also make the mark list.

Proposed system

The main objective of the online examination system is that it helps companies/institutions to conduct exams to any number of candidates at a time, in an automated manner. It reduces the time consumption and workload that exist in the current system of examination. It also helps in storing the record of each examination and the results are also stored in the system. This makes the searching of the records easier than the existing system.

Feasibility study

Feasibility is a measure of how beneficial the development of the information system will be to an organization. This is done by investigating the existing system in the area under investigation or generally ideas about a new system. It is a test of a system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources.

Three key considerations are involved in the feasibility analysis: economic, technical, and legal.

Technical feasibility

In examining Technical feasibility of the system, more importance is given to the hardware interaction part of the system. The assessments of technical feasibility centers on the existing system and to what extent it can support the proposed addition. This was based on an outline design of system requirements in turns of inputs, files, programs, procedures, and staff. It involves financial considerations to accommodate technical enhancements. Online_Examination being a web based application, it uses .Net framework, 800MHZ computer, 20 GB Hard disk.

Economic feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of a proposed system. It is more commonly known as cost benefit analysis, the procedure to determine the benefits and saving that are expected from a candidate system and compare them with costs. If the benefits outweigh costs then a decision is made to design and implement the system. Otherwise make alterations in the proposed system.

The innovation of the new system has much influence on the economical side of the company. Manuel system is highly cost driven due to the high labor costs. So if a company registers with the Online_Examination site, they can automate their day-to-day activities. Thus the system is economically feasible.

SOFTWARE REQUIREMENT SPECIFICATION

Functional Requirements

Since the Administrator and the student/user are the main target group of our software, we will only concern about some important functions for the admin and the user.

Administrator:-

- The administrator is the one who manipulates and maintains the system. He can enter into the system by entering login name and password
- That is, he is responsible for creating exams that include subject selection and assigning scores etc.
- Again, he can add questions to the database
- add new user to the database and issue a valid ID for the user.
- He is also responsible for sending the result to the email id provided by the user at the start of his registration.

Students/Users:-

- Can do the member registration
- After the registration , he will be issued with valid ID by the Administrator. The user can log into the system with this ID .

- After successfully login into the system, the user moves to the instruction web page where he will get instruction about the examination process.
- Then after clicking the start button the exam starts and timer also starts .In this manner, the user can take up the test and on clicking the submit button , he will get the result of that section immediately.

- He must get the test result to his email id provided at the start of the registration
- During the exam, he is allowed to go to the prevoious questions using a "previous" button. At the end system displays the initial web page.

Non Functional Requirements

It include the following interfaces

- User Interfaces
- Software Interfaces
- Hardware Interfaces

User Interfaces:-

The interface must be easy to understand. The user interface includes

- **screen formats/organization**: The introductory screen will be the first to be displayed which will allow the users to do the member registration.
- **window format/organization**: When the user chooses some other option, then the information pertaining to that choice will be displayed in a new

window which ensures multiple windows to be visible on the screen and the users can switch between them.

- **data format**: The data entered by the users will be alpha numeric.
- **end messages**: When there are some exceptions raising error like entering invalid details, then error messages will be displayed prompting the users to re-enter the details.

Hardware Interfaces:-

Server side hardware

- Hardware recommended by all the software needed.
- Communication hardware to serve client requests

Client side hardware

- Hardware recommended by respective client's operating system and web browser.
- Communication hardware to communicate the server.

Software Interfaces:-

Server side software

- Database tools: firebase.
- Compatible operating system: Windows XP

Client side software

- Web browser supporting JavaScript, refer Browser Compatibility

Communications Interfaces:-

http **-** Hypertext Transfer Protocol is a transaction oriented client/server protocol between web browser & a Web Server.

tcp/ip**:** Transmission Control Protocol/Internet Protocol, the suite of communication protocols used to connect hosts on the Internet. TCP/IP uses several protocols, the two main ones being TCP and IP.

Hardware Requirements

Processor	:	Pentium iv
Ram	:	256 mb
Hdd	:	2 gb
Monitor	:	svga color
Key board	:	normal
Mouse	:	normal

Software Requirements

Operating System	:	Windows XP
Front End	:	HTML
Back End	:	javascript

Web server : firebase

Browser : Google Chrome

3.5 Performance Requirements

- System should be able handle multiple users
- Database updating should follow transaction processing to avoid data inconsistency.

ARCHITECTURE AND DESIGN OF PROPOSED SYSTEM

System architecture

Design is the first step in the development phase for any engineered product or system. The design is the transition form the user oriented view to the programmer view. Design phase act as an edge between the software specification phase and the software development phase which satisfies the requirements. The system transforms a logical representation of what a given system is required

to be, into the physical specification. Design starts with the requirement specification and converts it into physical specification.

.System design is a creative art of inventing and developing inputs, databases, offline files, methods and procedures, for processing data to get meaningful output that satisfy the organization objectives. Through

the design phase consideration to the human factors, i.e., the inputs to the users will have on the system.

Some of the main factors that have to be noted using the design of the system are:

- Practicability

System must be capable of being operated over a long period of time and must have ease of use.

- Efficiency

Should make better use of resources available. Efficiency involves

Accuracy, timeliness and comprehensiveness of system output.

- Cost : Aim of minimum cost and better results
- Security : Physical security of data

Program Structure

Top-down programming is the opposite of bottom-up programming. It refers to a style of programming where an application is constructed starting with a high-level description of what it is supposed to do, and breaking the specification down into simpler and simpler pieces, until a level has been reached that corresponds to the primitives of the

programming language to be used. Top-down programming tends to generate modules that are based on functionality, usually in the form of functions or procedures. Typically, the high-level specification of the system states functionality. This high-level description is then refined to be a sequence or a loop of simpler functions or procedures, that are then themselves refined, etc. In this style of programming,

there is a great risk that implementation details of many data structures have to be shared between modules, and thus globally exposed. This in turn makes it tempting for other modules to use these implementation details, thereby creating unwanted dependencies .

Description for Components

There are four types of the modules

1: Student Module

2: Course Module

3: Exam Module

4: Administrator module

1. Student Module: - The student module contain another module

1.1 Registration Module: - Each of the students must first register with the software. For the registration part of the student will have to enter his details like name, address etc. and get a User Id from the software. Once the student register with software they are able to give the exam.

2. Course Module: - The course module contain another three module

2.1.Course Registration: -This modules contain all the information about different courses.

2.2.Question Entry: - This module contains all the question of different subjects. The questions are multiple types.

2.3.Mark Entry: - This module contains the mark details of different subjects.

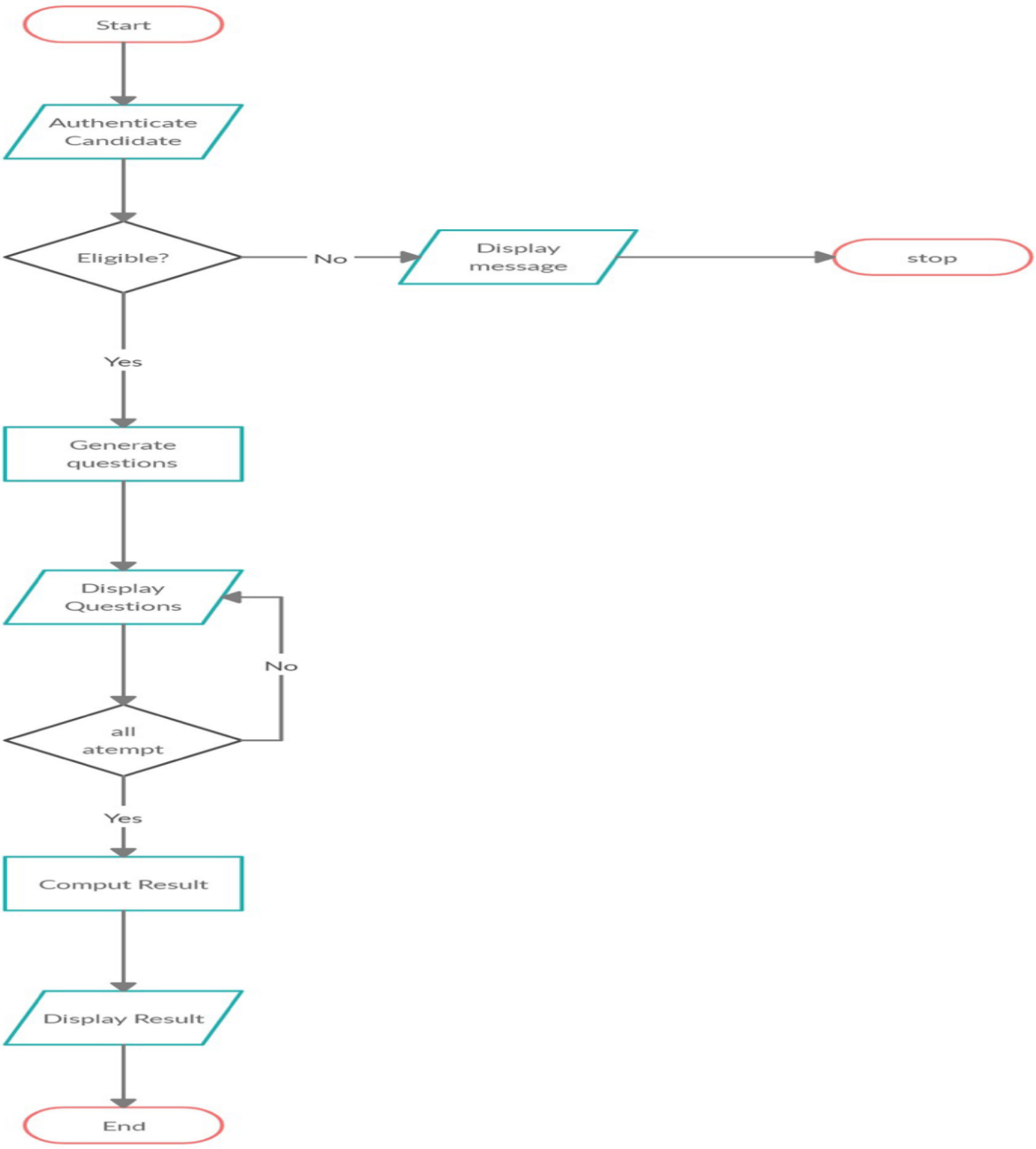
3. Exam Module: - This module is used for performing examination process. Time slot is allotted for exam. Two hour for each exam. Copy writing not possible in the exam. Result generation is also the part of this module. As the part of the result generation make the mark list. The

mark list contains Register number, name, course, subject, semester, mark, etc.

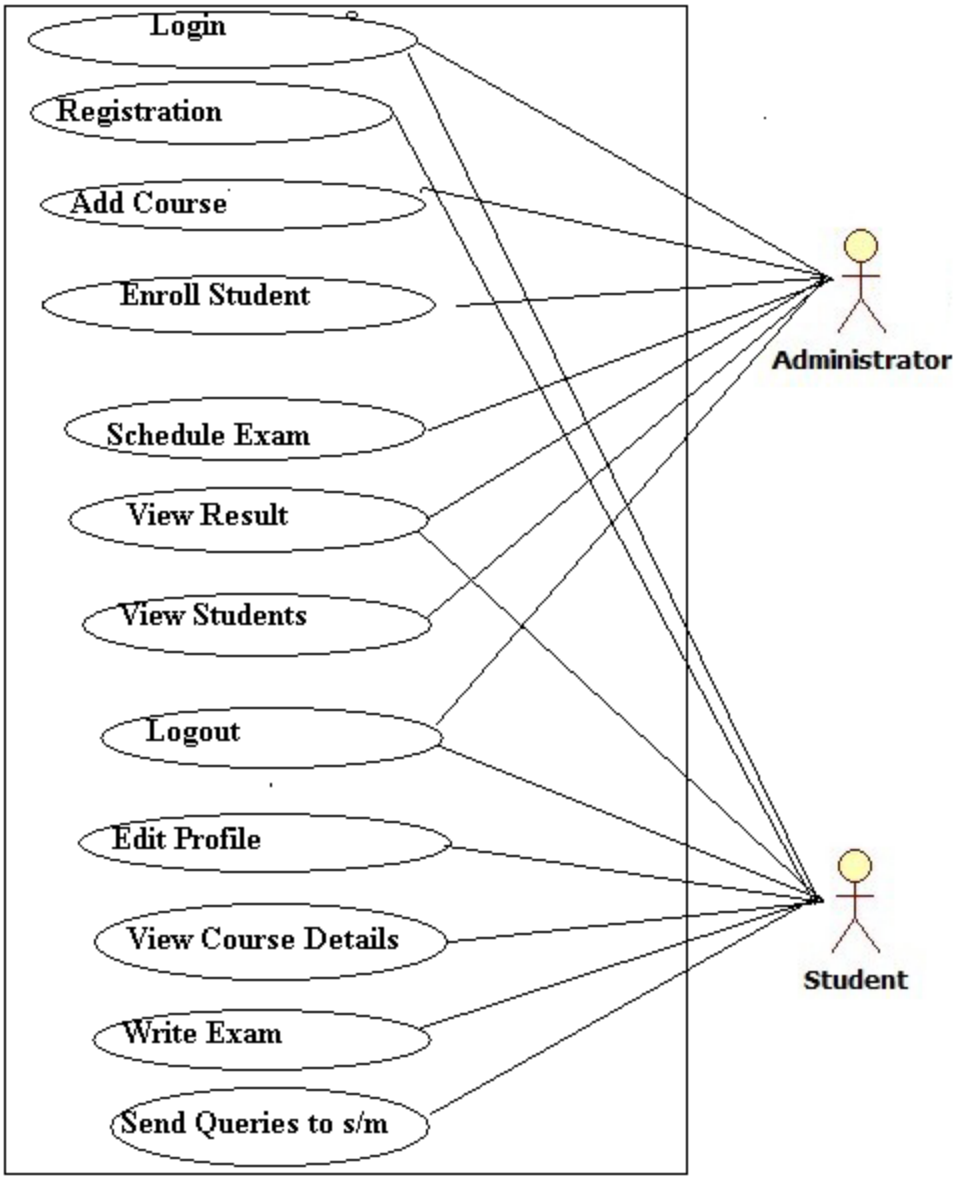
4. Administrators Module: - The module protected by user id and password.This is encrypted format. So Ordinary users of the software will not be permitted to enter this area of the software. The module will be focusing on the maintenance like Master Data entry operation.

Architectural diagrams/Data Flow Diagram/UML/

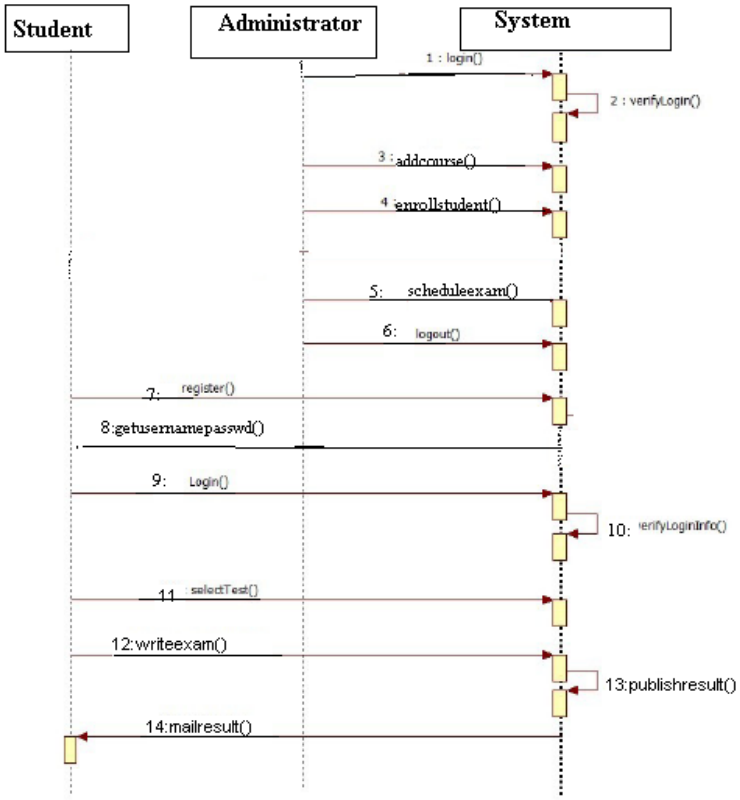
Flowchart:



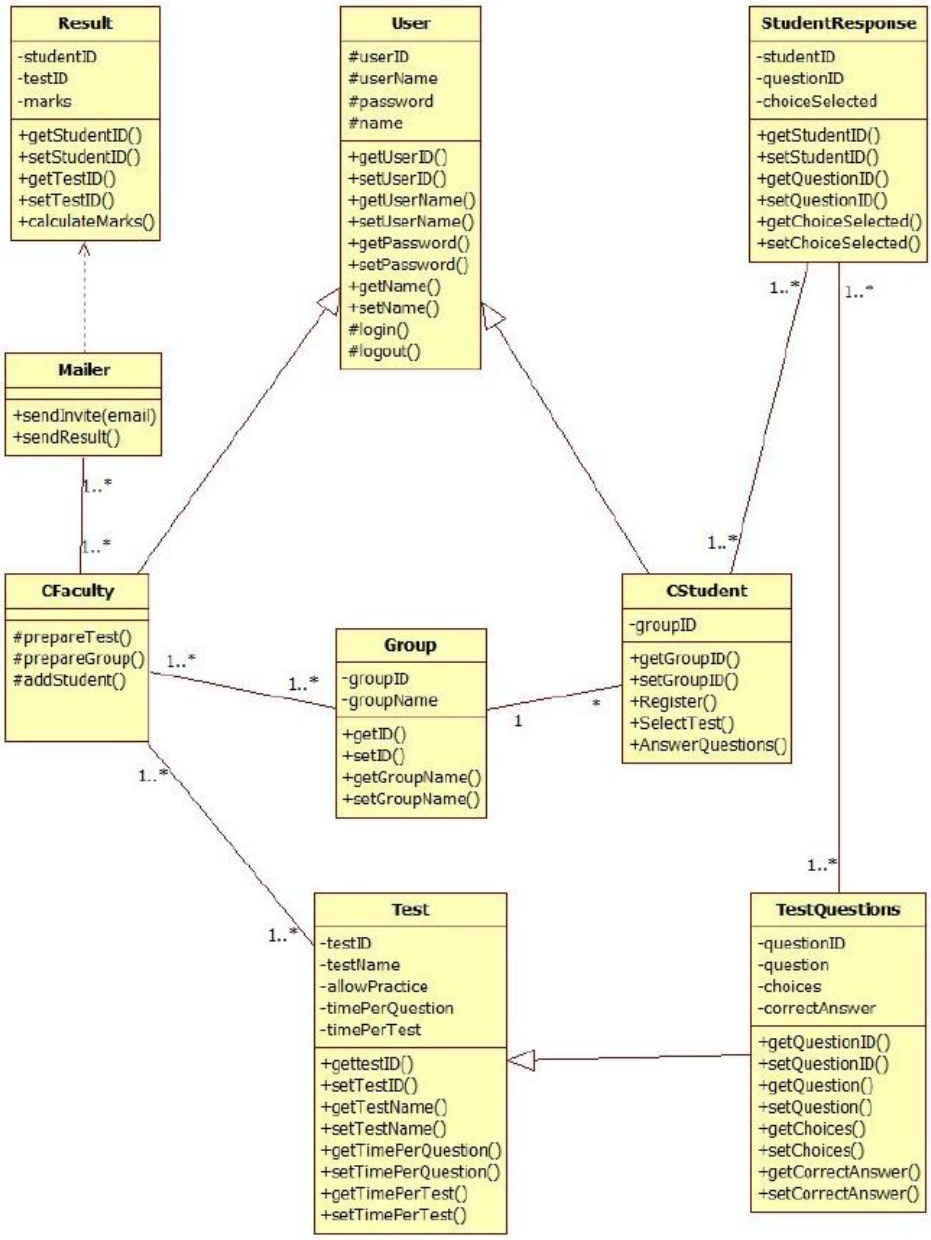
Use case diagram:



Sequence diagram:-



Class diagram:-


Result
-studentID
-testID
-marks
+getStudentID()
+setStudentID()
+getTestID()
+setTestID()
+calculateMarks()
User
#userID
#userName
#password
#name
+getUserID()
+setUserID()
+getUserName()
+setUserName()
+getPassword()
+setPassword()
+getName()
+setName()
#login()
#logout()
StudentResponse
-studentID
-questionID
-choiceSelected
+getStudentID()
+setStudentID()
+getQuestionID()
+setQuestionID()
+getChoiceSelected()
+setChoiceSelected()
Mailer
+sendInvite(email)
+sendResult()
CFaculty
#prepareTest()
#prepareGroup()
#addStudent()
Group
-groupID
-groupName
+getID()
+setID()
+getGroupName()
+setGroupName()
CStudent
-groupID
+getGroupID()
+setGroupID()
+Register()
+SelectTest()
+AnswerQuestions()
Test
-testID
-testName
-allowPractice
-timePerQuestion
-timePerTest
+gettestID()
+setTestID()
+getTestName()
+setTestName()
+getTimePerQuestion()
+setTimePerQuestion()
+getTimePerTest()
+setTimePerTest()
TestQuestions
-questionID
-question
-choices
-correctAnswer
+getQuestionID()
+setQuestionID()
+getQuestion()
+setQuestion()
+getChoices()
+setChoices()
+getCorrectAnswer()
+setCorrectAnswer()
1..*
1..*
1..*
1..*
1..*
1..*
1..*
1..*
1..*
1
*
1..*
1..*

IMPLEMENTATION

Introduction to technology

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users 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 change over, an evaluation, of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation.

An implementation co-ordination committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions are made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

Implementation is the final and important phase. This is the most critical stage in achieving a successful new system and in giving the users confidence that the new system will work is effective. The system can be implemented only after thorough testing. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

Sample code

The following are the html documents used for creating the online examination website.

Mainpage:

```
<!DOCTYPE html>

<html><head>

<style>

a:link, a:visited {

  background-color: #932FDC ;

  color: white;

  padding: 10px 20px;

  text-align: center;

  text-decoration: none;

  display: inline-block;

}

a:hover, a:active {

  background-color: red;

}

div {

  margin: 2px;

  border: 3px solid #239D64

   ;
```

```
}
f {
  text-decoration:none;
  color: #900C3F;
}
body {
  background-image:"au.png";
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: 50% 50%;
}
body {
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
}

body  {
  background-image: url("books.jpg");
  background-color: #cccccc;
background-repeat: no-repeat;
  background-size:100% 100%;
}
```

```
</style>

</head>

<body>

<center><img src="auemblem.png"></center>

<f><h1 style="font-size:2vw;"><center> Andhra University</center></h1></f>

<div><h2><center>ONLINE EXAMINATION</center></h2>

<ul>

    <h4><li>Now-a-days people are laking general awarness and social knowledge</li>

    <li>This test ,enables us to access the knowledge in the fields of general knowledge(GK) and Social Studies</li>

    <li>if you didnot register,click on "register" to register</li>

    <li>click on login if you have registered</li></h4>

</ul>

<center>

  <a href="register.html">Register</a>

  <a href="log.html">Login</a><br><br>

<script src="main.js"></script>

</div>
```

```
</body>
</html>
```

Register page:

```
<!DOCTYPE html>
<html>
<body>
 <style>
 body {font-family: Arial, Helvetica, sans-serif;}
 * {box-sizing: border-box}

 /* Full-width input fields */
 input[type=text], input[type=password] {
   width: 100%;
   padding: 15px;
   margin: 5px 0 22px 0;
   display: inline-block;
   border: none;
   background: #f1f1f1;
 }

 input[type=text]:focus, input[type=password]:focus {
   background-color: #ddd;
```

```
  outline: none;
}

hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}



/* Set a style for all buttons */
button {
  background-color: #3279E8;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
  opacity: 0.9;
}
```

```
button:hover {
  opacity:1;
}

/* Extra styles for the cancel button */
.cancelbtn {
  padding: 14px 20px;
  background-color:  #E04F49;
}

/* Float cancel and signup buttons and add an equal width */
.cancelbtn, .signupbtn {
  float: left;
  width: 50%;
}

/* Add padding to container elements */
.container {
  padding: 16px;
}

/* Clear floats */
```

```
 .clearfix::after {
  content: "";
  clear: both;
  display: table;
 }

 /* Change styles for cancel button and signup button on extra small screens
*/
 @media screen and (max-width: 300px) {
  .cancelbtn, .signupbtn {
     width: 100%;
  }

 </style>
 <h1>Registration form</h1>
 <p>Please fill in this form to create an account.</p>

 <label for="Name"><b>Name(User name)</b></label>
 <input type="text" id="Name" placeholder="Enter Your Name"
name="name" required>

 <label for="Father Name"><b>Father Name</b></label>
```

```
 <input type="text" id="Father Name" placeholder="Enter Father's Name" name="Father Name" required>


 <label for="Institution Name"><b>Institution Name</b></label>

 <input type="text" id="Institution" placeholder="Enter Institution" name="Institution" required>


 <b><b>Select your Course</b><select>

    <option value="course">B.E/B.Tech</option>

    <option value="course">M.E/M.Tech</option>

    <option value="course">MCA</option>

    <option value="course">MSC</option>

 </select><br><br>


 <b><b>Select your Year of Studying</b><select>

    <option value="year">1</option>

    <option value="year">2</option>

    <option value="year">3</option>

    <option value="year">4</option>

 </select><br><br>


 <label for="email"><b>Email</b></label>
```

```
  <input type="text" id="email" placeholder="Enter Your Email" name="email" required >


  <label for="psw"><b>Create password</b></label>

  <input type="password" id="password" placeholder="Enter Password" name="password" required >


  <label for="psw-repeat"><b>Re-enter Password</b></label>

  <input type="text" id="rpassword" placeholder="Re-Enter Your Password" name="rpassword" required >


  <button type="button" onclick=cancel() class="cancelbtn"><a href="file:///C:/Users/Madhu/Desktop/proj/exam/dyn/main.html">Cancel</a></button>

  <button type="button" onclick=register() class="signupbtn">register</button>

  <script src="https://www.gstatic.com/firebasejs/7.8.0/firebase-app.js"></script>

  <script src="https://www.gstatic.com/firebasejs/7.8.0/firebase-analytics.js"></script>

  <script src="https://www.gstatic.com/firebasejs/7.6.1/firebase-auth.js"></script>

  <script src="https://www.gstatic.com/firebasejs/7.6.1/firebase-database.js"></script>

  <script src="register.js"></script>
```

```
</body>

</html>
```

Login page:

```
<!DOCTYPE html>

<html>

<style>

body {font-family: Arial, Helvetica, sans-serif;}

* {box-sizing: border-box}


/* Full-width input fields */

input[type=text], input[type=password] {

  width: 100%;

  padding: 15px;

  margin: 5px 0 22px 0;

  display: inline-block;

  border: none;

  background: #f1f1f1;

}


input[type=text]:focus, input[type=password]:focus {

  background-color: #ddd;

  outline: none;
```

```
}

hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}

/* Set a style for all buttons */
button {
  background-color: #5488D5;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
  opacity: 0.9;
}

button:hover {
  opacity:1;
}
```

```
/* Extra styles for the cancel button */
.cancelbtn {
  padding: 14px 20px;
  background-color: #CA8262;
}

/* Float cancel and signup buttons and add an equal width */
.cancelbtn, .signupbtn {
  float: left;
  width: 50%;
}

/* Add padding to container elements */
.container {
  padding: 16px;
}

/* Clear floats */
.clearfix::after {
  content: "";
  clear: both;
```

```
  display: table;
}

/* Change styles for cancel button and signup button on extra small screens */
@media screen and (max-width: 300px) {
  .cancelbtn, .signupbtn {
     width: 100%;
  }
}
</style>
<body>
   <h1>Log in</h1>
   <p>enter your details</p>
   <hr>

   <label for="email"><b>UserName(enter your name)</b></label>
   <input type="text" id="email" placeholder="Enter name" name="email" required>

   <label for="psw"><b>Password</b></label>
   <input type="password" id= "password" placeholder="Enter Password" name="psw" required>
```

```
    <button type="submit" onclick=login() class="signupbtn">Login(enter the test)

  <button type="button" class="cancelbtn">Cancel</button>

<script src="https://www.gstatic.com/firebasejs/7.8.0/firebase-app.js"></script>

<script src="https://www.gstatic.com/firebasejs/7.8.0/firebase-analytics.js"></script>

<script src="https://www.gstatic.com/firebasejs/7.6.1/firebase-auth.js"></script>

<script src="https://www.gstatic.com/firebasejs/7.6.1/firebase-database.js"></script>

<script src="log.js"></script>

</body>

</html>
```

Quiz page:

```
<html>

<head><title>Online Examination System</title></head>

<body>

  <style>

    body {font-family: Arial, Helvetica, sans-serif;}

    * {box-sizing: border-box}

    .submitbtn {
```

```
    background-color: #3279E8;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    cursor: pointer;
    width: 100%;
    opacity: 0.9;
  }
  button:hover {
    opacity:1;
  }
  </style>
 <script src="https://www.gstatic.com/firebasejs/7.8.2/firebase-app.js"></script>
 <script src="https://www.gstatic.com/firebasejs/7.8.0/firebase-app.js"></script>
 <script src="https://www.gstatic.com/firebasejs/7.8.0/firebase-analytics.js"></script>
 <script src="https://www.gstatic.com/firebasejs/7.6.1/firebase-auth.js"></script>
 <script src="https://www.gstatic.com/firebasejs/7.6.1/firebase-database.js"></script>
```

```
 <!-- The core Firebase JS SDK is always required and must be listed first -->

<!-- TODO: Add SDKs for Firebase products that you want to use
    https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/7.8.2/firebase-analytics.js"></script>

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyC1Zf_VsRWkrlrxLQDNYBJ1Ait1UCcqhIs",
    authDomain: "online-examination-syste-b1886.firebaseapp.com",
    databaseURL: "https://online-examination-syste-b1886.firebaseio.com",
    projectId: "online-examination-syste-b1886",
    storageBucket: "online-examination-syste-b1886.appspot.com",
    messagingSenderId: "642625682916",
    appId: "1:642625682916:web:722a09d9fea544bd4a1bf1",
    measurementId: "G-98L1EW8YGE"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
```

```
  firebase.analytics();

</script>

   <script src = "quiz.js"></script>

   <button type = "button" onclick = submit()
class="submitbtn">Submit</button>

   <h1 id = "scorecard"></h1>

   </body>

   </html>
```

Administrator page:

```
<html>

<head><title>Online Examination System</title></head>

<body>

  <script src="https://www.gstatic.com/firebasejs/7.8.2/firebase-
app.js"></script>

  <script src="https://www.gstatic.com/firebasejs/7.8.0/firebase-app.js"
></script>

  <script src="https://www.gstatic.com/firebasejs/7.8.0/firebase-
analytics.js"></script>

  <script src="https://www.gstatic.com/firebasejs/7.6.1/firebase-
auth.js"></script>

  <script src="https://www.gstatic.com/firebasejs/7.6.1/firebase-
database.js"></script>


  <!-- The core Firebase JS SDK is always required and must be listed first -->
```

```
<!-- TODO: Add SDKs for Firebase products that you want to use
    https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/7.8.2/firebase-analytics.js"></script>

<script>

  var firebaseConfig = {
    apiKey: "AIzaSyC1Zf_VsRWkrlrxLQDNYBJ1Ait1UCcqhIs",
    authDomain: "online-examination-syste-b1886.firebaseapp.com",
    databaseURL: "https://online-examination-syste-b1886.firebaseio.com",
    projectId: "online-examination-syste-b1886",
    storageBucket: "online-examination-syste-b1886.appspot.com",
    messagingSenderId: "642625682916",
    appId: "1:642625682916:web:722a09d9fea544bd4a1bf1",
    measurementId: "G-98L1EW8YGE"
  };

  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
```

```
  </script>

  <script src="administrator.js"></script>

  <button type = "button" onclick ="logout()" id="logout" style="position:
absolute; right:10px">Log Out</button>

  <div class="main">

  <table border="0">

    <font color="white">

   <tr><td><font color="#6D214F"><b>Question-</b></font></td><td> <input
type = "text" id = "q" ></body></td></tr>

    <tr><td><font color="#6D214F"><b>Option A-</b></font></td><td> <input
type = "text" id = "a" ></body></td></tr>

   <tr><td><font color="#6D214F"><b> Option B-</b></font></td><td><input
type = "text" id = "b" ></body></td></tr>

   <tr><td><font color="#6D214F"><b>Option C-</b></font></td><td><input
type = "text" id = "c" ></body></td></tr>

   <tr><td><font color="#6D214F"><b> Option D-</b></font></td><td> <input
type = "text" id = "d" ></body></td></tr>

   <tr><td> <font color="#6D214F"><b>Answer-</b></font></td><td>  <input
type = "text" id = "ans" ></body></td></tr>

  <tr><td></td><td><button  type = "button" onclick = upload()
id="update">Upload</button></td></tr>


</table
```

```
 <button type = "button" onclick = showscores() id="score">Show
Scores</button>


 </div>
 </body>
 </html>
```

The following are the JS files used for creating the online examination website.

Mainpage:

```
// Your web app's Firebase configuration
credentials();
signout();


function credentials(){
 var firebaseConfig = {
   apiKey: "AIzaSyC1Zf_VsRWkrlrxLQDNYBJ1Ait1UCcqhIs",
   authDomain: "online-examination-syste-b1886.firebaseapp.com",
   databaseURL: "https://online-examination-syste-b1886.firebaseio.com",
   projectId: "online-examination-syste-b1886",
   storageBucket: "online-examination-syste-b1886.appspot.com",
   messagingSenderId: "642625682916",
```

```
  appId: "1:642625682916:web:722a09d9fea544bd4a1bf1",
  measurementId: "G-98L1EW8YGE"
 };
 // Initialize Firebase
 firebase.initializeApp(firebaseConfig);
 firebase.analytics();
}

function signout(){
 firebase.auth().signOut().then(function() {
  // Sign-out successful.
 }).catch(function(error) {
  // An error happened.
 });
}

 firebase.auth().onAuthStateChanged(function(user) {
  if (user) {
   window.location.href="register.html";
  } else {
   // No user is signed in.
  }
```

```
});
```

Register page:

```
// Your web app's Firebase configuration
credentials();
signout();

var name,fname,iname;

function credentials(){
 var firebaseConfig = {
   apiKey: "AIzaSyC1Zf_VsRWkrlrxLQDNYBJ1Ait1UCcqhIs",
   authDomain: "online-examination-syste-b1886.firebaseapp.com",
   databaseURL: "https://online-examination-syste-b1886.firebaseio.com",
   projectId: "online-examination-syste-b1886",
   storageBucket: "online-examination-syste-b1886.appspot.com",
   messagingSenderId: "642625682916",
   appId: "1:642625682916:web:722a09d9fea544bd4a1bf1",
   measurementId: "G-98L1EW8YGE"
 };
 // Initialize Firebase
 firebase.initializeApp(firebaseConfig);
```

```
 firebase.analytics();
}

function signout(){
 firebase.auth().signOut().then(function() {
   // Sign-out successful.
 }).catch(function(error) {
   // An error happened.
 });
}
firebase.auth().onAuthStateChanged(function(user) {
  if (user) {
    var uid = user.uid;
    firebase.database().ref('users/'+uid).set({
      Name: name,
      FatherName: fname,
      INSTITUTION : iname
    }, function(error) {
      if (error) {
        alert("Data not Saved");
        // The write failed...
      } else {
```

```
      alert("Now, proceed to login");
    }
   });
  } else {
   // No user is signed in.
  }
});

function register() {
   name = document.getElementById("Name").value;
   fname = document.getElementById("Father Name").value;
   iname = document.getElementById("Institution").value;
  var email=document.getElementById("email").value;
  var password=document.getElementById("password").value;
  var rpassword=document.getElementById("rpassword").value;

  if(password==rpassword){
   alert("Registered Succesfully");
   firebase.auth().setPersistence(firebase.auth.Auth.Persistence.LOCAL)
   .then(function() {
     return firebase.auth().createUserWithEmailAndPassword(email,password);
   })
```

```
    .catch(function(error) {
      // Handle Errors here.
      var errorCode = error.code;
      var errorMessage = error.message;
      alert(errorMessage);
    });
     }
     else {
     alert("Password mismatch");
    }
  }
```

Login page:

```
credentials();
signout();


function credentials(){
 var firebaseConfig = {
   apiKey: "AIzaSyC1Zf_VsRWkrlrxLQDNYBJ1Ait1UCcqhIs",
   authDomain: "online-examination-syste-b1886.firebaseapp.com",
   databaseURL: "https://online-examination-syste-b1886.firebaseio.com",
   projectId: "online-examination-syste-b1886",
   storageBucket: "online-examination-syste-b1886.appspot.com",
```

```
  messagingSenderId: "642625682916",
  appId: "1:642625682916:web:722a09d9fea544bd4a1bf1",
  measurementId: "G-98L1EW8YGE"
 };
 // Initialize Firebase
 firebase.initializeApp(firebaseConfig);
 firebase.analytics();
}

function signout(){
 firebase.auth().signOut().then(function() {
  // Sign-out successful.
 }).catch(function(error) {
  // An error happened.
 });
}
function login(){

  var email=document.getElementById("email").value;
  var password=document.getElementById("password").value;
```

```
firebase.auth().signInWithEmailAndPassword(email,
password).catch(function(error) {
  // Handle Errors here.
  var errorCode = error.code;
  var errorMessage = error.message;
  // ...
});
firebase.auth().onAuthStateChanged(function(user) {
 if (user) {
   window.location.href="quiz.html";
 } else {
   // No user is signed in.
 }
});
)
```

TESTING

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation

commences. Testing is vital to the success of the system. Testing is the process of executing a program with the explicit intention of finding errors that is making the program fail. The tester may analysts, programmer or a specialist trained for software testing, is actually trying to make the program fail. Analysts know that an effective testing program does not guarantee system reliability. Therefore reliability must be designed into the system.

Unit Testing

In unit testing we have to test the programs making up the system. For this reason unit testing is sometimes called as the Program testing. The software units in a system are modules and routines that are assembled and integrated to perform a specific function.

Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained with in the module alone. Unit testing can be performed from the bottom up, starting with the lowest level modules and proceeding one at a time. Unit testing is done for each module in Online_Examination. This ensures that the value we enter match with the data type and within the specified limits.

Integration Testing

Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing for conducting tests to uncover errors associated within the

interface. The objective is to take unit tested modules and build a program

structure. All the modules are combined and tested as a whole. Here correction is difficult because the vast expenses of the entire program complicate the isolation of causes. Thus in the integration testing step, all the errors are corrected for the next testing steps. In Online_Examination each module is integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

Validation Testing

At the culmination of integration testing, software is completely assembled as a package; interfacing errors have been recovered and

corrected and a final series of a software tests-validation tests begin.

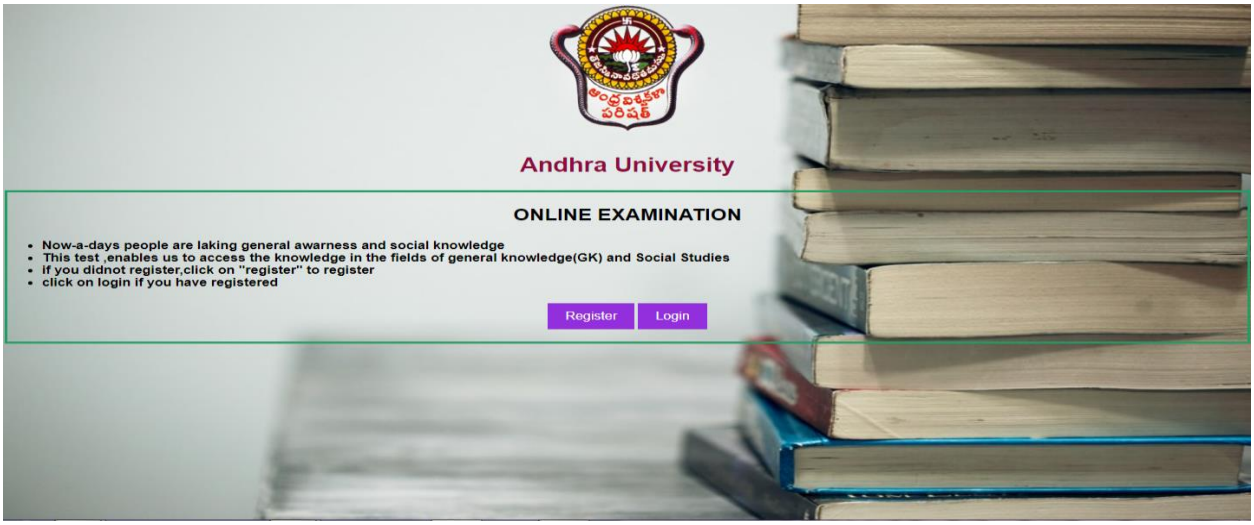
Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

In validation testing if user wants to enter the numeric value he can only enter the numeric value not the text value. For e.g.: in phone number field user can only enter numeric value to it. The system is user friendly with user guide and messages to explain further procedures. An attempt has been made to perfect the process by incorporating validation at each level.

RESULTS AND DISCUSSION

Screenshots of the website:

Main page:



Registration page:

Registration form

Please fill in this form to create an account.

Name(User name)

peter

Father Name

john

Institution Name

andhra university

Select your Course B.E/B.Tech

Select your Year of Studying 3

Email

peterjohn@gmail.com

Create password

••••••••••

Re-enter Password

peterpeter

Cancel | register

Login page:

Log in

enter your details

UserName(enter your name)

peterjohn@gmail.com

Password

••••••••••

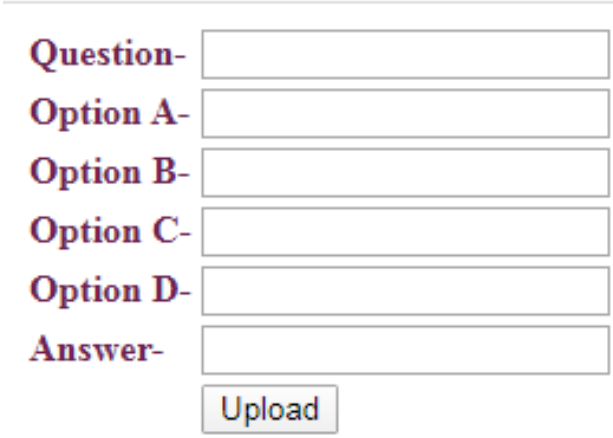
Login(enter the test) | Cancel

Quiz:

Submit

1.	Who was the first Indian in Space?	Vikram Ambalal Ravish Malhotra Rakesh Sharma None Of The Above
2.	Who Built The Jama Masjid?	Jahangir Akbar Imam Bukhari Shah Jahan
3.	Who was the first Indian woman in Space?	Kalpana Chawla Sunita Williams Koneru Humpy None Of The Above
4.	Who wrote the Indian National Anthem?	Bakim Chandra Chatterji Rabindranath Tagore Swami Vivekanand None of the above
5.	Who was the first Indian Scientist to win a Nobel Prize?	CV Raman Amartya Sen Hargobind Khorana Subramanian Chandrasekhar
6.	1+2?	34 4 5 3

Administrator page:



Result:

Adding a question by the administrator:

Updation of the quiz:

Conclusion

Online_Examination has been developed and the system was tested with proper data. The system results in regular timing preparation of the required output. In comparison with the manual system, the benefit under a computer system considerable in to saving of manpower, working hour and efforts.

It can observe that the information required can be obtained with ease and accuracy in the computerized system. The user with minimum knowledge about computer can be able operate the system easily. Online massage has been provided to help the user to take necessary, correct action while using the system. Various validation techniques have been used to implement accuracy of data in all formats of input. The system has produced all the report required by the management .

This software can be used by any institute as it can be modified easily; additional features can be added without interrupting the normal functioning of the system.