##### A Project report on

##### INSTANT QUIZ CONTEST

###### A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

**Bachelor of Technology**

**IN**

**Computer Science and Engineering**

Submitted by

(Student Name) (Roll No)

S.MOULIKA (19H51A05B8)

R.ROHITH REDDY (19H51A05E8)

M.LAKSHMI GANESH REDDY (19H51A05H4)

Under the esteemed guidance of

**(MR.B TULASI DASU)**

**(ASSISTANT PROFESSOR)**



**Department of Computer Science and Engineering**

**CMR COLLEGE OF ENGINEERING AND TECHNOLOGY**

(An Autonomous Institution under UGC & JNTUH , Approved by AICTE, Permanently Affiliated to JNTUH, Accredited by NBA.)

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2019- 2023

**CMR COLLEGE OF ENGINEERING & TECHNOLOGY**

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



#### CERTIFICATE

This is to certify that the Mini Project-1 report entitled “**INSTANT QUIZ CONTEST”** being submitted by **S. MOULIKA (19H51A05B8), R. ROHITH REDDY (19H51A05E8), M. LAKSHMI GANESH REDDY (19H51A05H4)**in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of Bonafide work carried out his/her under my guidance and supervision.

###### The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

**MR.B. TULASI DASU DR. K. VIJAYA KUMAR**

**ASSISTANT PROFESSOR PROFESSOR AND HOD**

**Dept. of CSE Dept. of CSE**



##### ACKNOWLEDGEMENT

We are highly indebted and grateful to our guide MR.B TULASI DASU, ASSISTANT PROFESSOR for his excellent guidance and constant encouragement throughout for the successful completion of the Project.

We are obliged and grateful to thank Dr. K VIJAYA KUMAR, Head of the Department of Computer Science and Engineering, CMRCET, for his cooperation in all respects.

We would like to thank Major Dr.VA NARAYANA, Principal, for his support in the course of this project work.

We would like to thank Sri Ch. GOPAL REDDY , Secretary & Correspondent of CMRCET, for his cooperation in all respects during the course.

Finally we would like to thank all teaching & non- teaching staff members of the department, for their cooperation and support throughout the duration of our course.

Ultimately we own all our success to our beloved parents, whose vision, love and inspiration has made us to reach out for these glories.



**DECLARATION**

We hereby declare that results embodied in this Report of Project on **“INSTANT QUIZ CONTEST”** are from work carried out by using partial fulfillment of the requirements for the award of B. Tech degree. We have not submitted this report to any other university/institute for the award of any other degree.

**NAME SIGNATURE**

S.MOULIKA (19H51A05B8)

R.ROHITH REDDY (19H51A05E8)

M.LAKSHMI GANESH REDDY (19H51A05H4)



**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTERS** | | **DESCRIPTION** | **PAGE No** |
|  |  | **Abstract** | 3 |
| **1** |  | **Introduction** | 4 |
|  | 1.1 | Objective | 4 |
| **2** |  | **Existing solutions** | 5 |
| **3** |  | **Proposed System** | 6 |
|  | 3.1 | Proposed solution | 6 |
|  | 3.2 | Description | 7 |
|  | 3.3 | Requirements | 7 |
|  | 3.4 | Modules in system | 8 |
|  | 3.5 | How it works | 9 |
| **4** |  | **Designing** | 10 |
| **5** |  | **Results and Discussions** | 11-16 |
|  | 5.1 | Advantages | 17 |
|  | 5.2 | Future scope | 17 |
|  | 5.3 | Execution code | 18-35 |
| **6** |  | **Conclusion and Reference** | 36 |
|  | 6.1 | Conclusion | 36 |
|  | 6.2 | References | 36 |

**ABSTRACT**

* In this pandemic most of the exams are conducting in the form of quiz through online mode. This program asks the questions to the students one after another about the subject which we inserted in program.
* Each correct answer will score a point. At the end of the quiz, the program will reveal the students final score.
* The score is stored in a variable during the quiz. Once all the questions have been answered, the quiz ends. Moreover, This project can also useful for the faculty to send the queries to the students as like they using google forms.

1. **INTRODUCTION**

The ‘INSTANT QUIZ CONTEST’ project will be developed to overcome the time consuming problem of manual system. Apart from that in current system, checking the answer sheets after taking test, waste the examiners time, so this application will check the correct answer and save the examiner time and carry the examination in an effective manner. The users which are use this system don’t need to high computing knowledge and also system will inform them while entering invalid data.

The aim of this project is to computerized the existing manual system and help the examiners to save their valuable time and important data. Apart from this, data which are exist in this system, will exist for long period of time and will be easy accessible. This project helps the examiners to manage their services in a good way and provide a better service to their users.

The objective of this project is to manage the details of students, examinations, marks, courses and papers in a good manner. The performance of the application will be fully control by administrator and administrator can guaranty any one to access. The project will reduce the manual process in managing examinations and all issues regarding that.

To design and implement this project we plan that the project support to different types of users apart from its administrative part. When project is run for the first time it allowed the user to select as who he/she wants to login in the system. Project support login as teacher and login as student. If a user who is student, try to login as teacher system will not allow him and vice versa. User who add as teacher in system will be able to punch test and questions to system and also will be able to observe the result of the student which attempt tests.

User who login to system as student will be able to login to test and attempt questions depend on the time. After attempting the test and submitting that user will receive a message that you have attempt the test successfully and they quiz will give the score of the student and if the user tries to attempt the same test, system will allow him/her but every thing will be reloaded and the process will begin from starting.

**OBJECTIVE:**

To build a webpage to conduct the instant quiz contest to test the grasping power of students or to know their knowledge over a particular topic.

**2.EXISTING SOLUTIONS**

**i. A Quiz Game with Python:**

The Quiz game asks the player questions about animals. They have three chances to answer each question you don’t want to take the quiz too difficult. Each correct answer will score a point. At the end of the game, the program will reveal the player’s final score. Once all the questions have been answered, the game ends.

### Drawbacks in existing system

* User can give the answers multiple times.
* It is not safe and the data doesn’t secure.
* Results all not available instantly.

**ii. PYTHON-MCQ QUIZ GAME USING TKINTER:**

Here we will be developing a simple multiple-choice quiz in python with GUI. We will be creating a multiple choice quiz in Python with Tkinter. Here we will use GUI for applications.

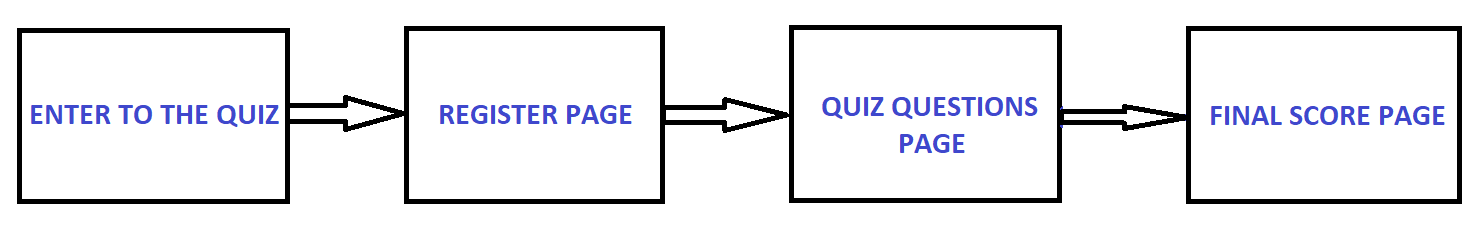
### Drawbacks in existing system

* A lot of effort is taken to conduct exams.
* Results will not appear directly after given input.
* It develops the exam anxiety among test takes.

**3. PROPOSED SYSTEM**

## PROPOSED SOLUTION:

The main aim of the project is to computerized the existing manual system & help the examiners to save their valuable time & important data. This project helps the examines to manage their services in a good way and provide a better services to their users. This quiz contest can remind the user about the time remaining for each question. Here the users can instantly know the correct answer after giving their input. This works more efficiently and with more clarity.



**DESCRIPTION:**

The proposed solution gives the users more clarity on the quiz and it will focus on the student concentration. For every question there will be timer and it will help the student to focus on the quiz and the user able to know the time remaining for each and every question and they can easily focus on their quiz.

The objective of this project is to manage the details of students, examinations, marks, courses and papers in a good manner. The performance of the application will be fully control by administrator and administrator can guaranty any one to access. The project will reduce the manual process in managing examinations and all issues regarding that.

# SYSTEM REQUIREMENTS:

**HARDWARE REQUIREMENTS:**

PROCESSOR : DUAL CORE 2 DUOS

RAM : 4 GB DD RAM

HARD DISK : 250 GB

**SOFTWARE REQUIREMENTS:**

OPERATING SYSTEM : WINDOWS 7/8/10

PLATFORM : SUBLIME TEXT EDITOR

PROGRAMMING LANGUAGE : HTML,CSS,JAVA SCRIPT,PHP

FRONT END : SUBLIME TEXT EDITOR

**MODULES IN SYSTEM:**

The **Quiz** activity module allows the teacher to design and build quizzes consisting of a large variety of [Question types](https://docs.moodle.org/25/en/Question_types), including multiple choice, true-false, and short answer questions. These questions are kept in the [Question bank](https://docs.moodle.org/25/en/Question_bank) and can be re-used in different quizzes.

* Feedback about performance and self-assessment are important parts of a learning environment. There are several ways to give feedback to students: on each question or overall. The quiz module can display feedback and scores at different times during the quiz, using the review options in the [Quiz settings](https://docs.moodle.org/25/en/Quiz_settings).
* A wide variety of [Quiz reports](https://docs.moodle.org/25/en/Quiz_reports) (in addition to [Grades](https://docs.moodle.org/25/en/Grades)) are available for use by the teacher. Quiz reports not only can focus on a single student's attempt to answer each question, but also can perform a robust item analysis of a question's validity based upon aggregated student responses.
* A single quiz can automatically select random and/or specific questions from different [categories](https://docs.moodle.org/25/en/Question_bank) of questions.
* There are different options for [scoring (marks-grades)](https://docs.moodle.org/25/en/Editing_a_quiz) individual questions in a specific quiz, [grading attempts](https://docs.moodle.org/25/en/Adding/updating_a_quiz#Grades_section) for a quiz and each [question type](https://docs.moodle.org/25/en/Question_types).
* The teacher can [choose](https://docs.moodle.org/25/en/Quiz_settings) how questions behave during the quiz. It can be like a classic test, where the student gets no feedback while attempting the test, only later. Alternatively, Moodle can reveal the grades and/or feedback to the student during the quiz, and perhaps even give them another chance to answer the question (for fewer marks) having read the feedback.

**HOW IT WORKS:**

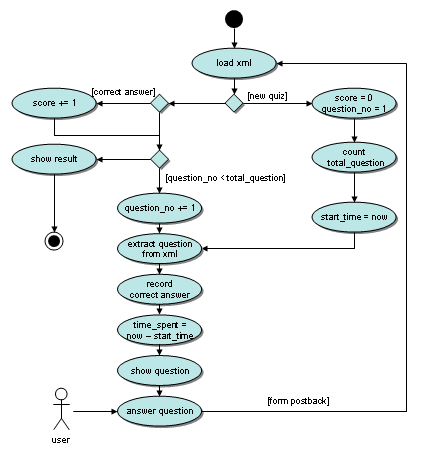
In this program [instant quiz contest], there are three layers or boxes, and these boxes shown one by one on a particular button clicked. At first, on the webpage, there is shown a button labelled as "ENTER TO THE QUIZ" and when you clicked on that button, then the info box appears with popup animation. In this info box, there are some rules of the quiz and two buttons labelled as "QUIT" and "NEXT". When you clicked on the Quit button, the info box will be hidden but when you clicked on the Next button, then the Quiz Box appears.

In the Quiz Box, there is a header with a title on the left side and a timer box on the right side. This timer starts decrement from 30 to 0 sec and there is also shown a timeline indicator that is sliding from the left to right side according to the timer. If the user selects an option between 30 to 0 sec, the timer will be stopped and all available options will be disabled. If the user selected option is correct, the selected option colour, background colour changed to green and there is also shown the tick icon to inform the user that the selected answer is correct. If the user selects an option that is incorrect, the selected option colour, background-colour changed to red and there is shown the cross icon to inform the user that the selected option is incorrect and the correct option will be automatically selected.

If the user doesn't select an option between 30 to 0 sec, the timer will be stopped once it comes in 0 and the correct option of that question will be selected automatically. After that, there is the next button to show the next question, and there is a total of five questions on this Quiz. In the end, the result box will be appeared and shown the user score and two buttons [Try again, Exit], if the user clicked on the Try again button, the quiz will again start with the number 1 question, and the score of the user will be 0 but if the user clicked on the Exit quiz button, the current window will be reloaded and the quiz starts from the begin.

**4. DESIGNING**

Flow chart for the quiz process:



Steps to implement the instant quiz contest web application using html, css and java script.

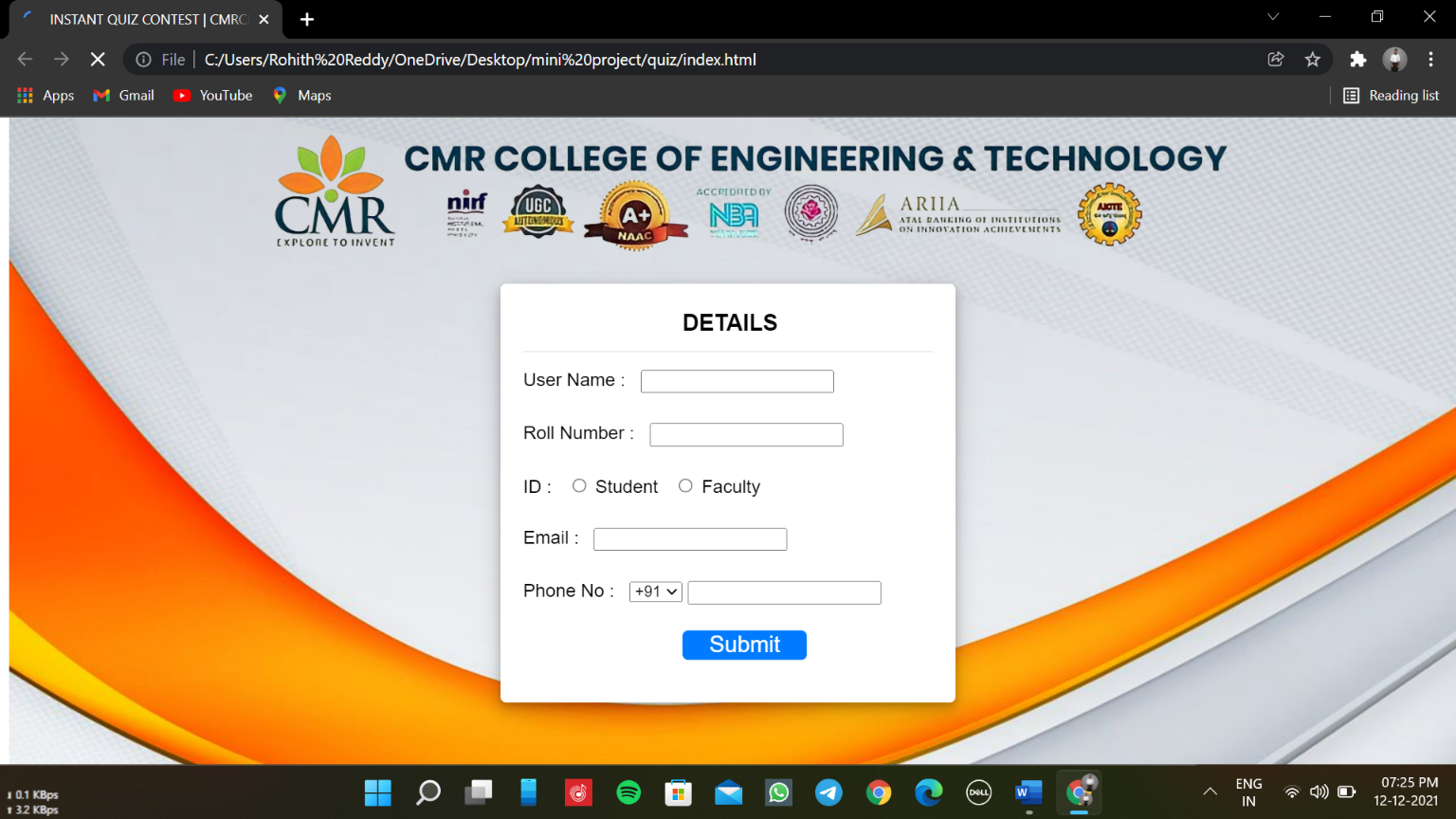
**5. RESULTS AND DISCUSSIONS:**

**WORKING PROTOTYPE IMAGES:**

**A screenshot of a computer

Description automatically generated with medium confidence**

At first, on the webpage, there is shown a button labelled as "ENTER TO THE QUIZ" and when you clicked on that button,



Then the “DETAILS” window will appear in that, details will be asked named as User Name, Roll Number, ID, Email, Phone No. Then after filling the details when you click on submit button then,

Graphical user interface, text

Description automatically generated

The info box appears with popup animation. In this info box, there are some “RULES AND REGULAATIONS” of the quiz and two buttons labelled as "QUIT" and "NEXT".

When you clicked on the Quit button, the info box will be hidden but when you clicked on the Next button, then the Quiz Box appears.

Graphical user interface, application

Description automatically generated

In the Quiz Box, there is a header with a title on the left side and a timer box on the right side. This timer starts decrement from 30 to 0 sec and there is also shown a timeline indicator that is sliding from the left to right side according to the timer.

If the user selects an option between 30 to 0 sec, the timer will be stopped and all available options will be disabled. If the user selected option is correct, the selected option colour, background colour changed to green and there is also shown the tick icon to inform the user that the selected answer is correct. Then,

Graphical user interface, application

Description automatically generated

If the user selects an option that is incorrect, the selected option colour, background-colour changed to red and there is shown the cross icon to inform the user that the selected option is incorrect and the correct option will be automatically selected.

If the user doesn't select an option between 30 to 0 sec, the timer will be stopped once it comes in 0 and the correct option of that question will be selected automatically. After that, there is the next question button to show the next question.

Graphical user interface, application

Description automatically generated

In the end, the result box will be appeared and shown the user score and two buttons [Try again, Exit Quiz], if the user clicked on the Try again button, the quiz will again start with the number 1 question, and the score of the user will be 0 but if the user clicked on the Exit quiz button, the current window will be reloaded and the quiz starts from the begin.

**ADVANTAGES OF PROPOSED SYSTEM :**

* This proposed solution gives more clarity for the user to attend the quiz.
* It gives instant solutions for the particular question.
* We can save more time.
* Brings competitive spirit among the students.
* Very much useful during these pandemic situation to know the student status.
* Timer will be added for every question so, the user can easily focus on the quiz.
* This proposed solution is more secure.

**FUTURE SCOPE:**

* Plays a vital role in pandemic situations.
* This project is only for MCQS test but in future we can extend it to support subjective type of questions with more functionality.
* We can able to add administrative part on it which able the system to delete test, add user, delete user and so on graphically via the web.
* This application is used in both educational institutions as well as corporate world.
* Can be used anywhere any time as it is a web based application.

**EXECUTION CODE:**

**HTML CODE:**

<!DOCTYPE html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>INSTANT QUIZ CONTEST | CMRCET</title>

<link rel="stylesheet" href="style.css">

<!-- FontAweome CDN Link for Icons -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>

</head>

<body>

<div class="xy">

<img src="FRONT.PNG" height="35%" ></div>

</div>

<div class="qlogo">

<img src="QUIZ LOGO.PNG" height="35%" ></div>

<div class="back">

<img src="BACK.JPG" height="35%" ></div>

</div>

<!-- start Quiz button -->

<div class="start\_btn"><button>ENTER TO THE QUIZ</button></div>

<div class="register\_btn">

<div class="register\_header"><span>DETAILS</span></div>

<p class="register\_name">User Name : &nbsp <input type="text" name="User Name" required></p>

<p>Roll Number : &nbsp <input type="text" name="Roll Number" required></p>

<p>ID : &nbsp <input type="radio" name="student" required> Student

&nbsp <input type="radio" name="student" required> Faculty</p>

<tr>

<p>Email : &nbsp <input type="email" name="email" required></p>

</tr>

<p>Phone No : &nbsp <select>

<option>+91</option>

<option>+92</option>

<option>+93</option>

</select>

<input type="phone" required>

</p>

<div class="register1\_btn">

<p><input type="submit" value=" Submit "></p>

</div>

</div>

<!-- Info Box -->

<div class="info\_box">

<div class="info-title"><span>RULES AND REGULATIONS of this Quiz</span></div>

<div class="info-list">

<div class="info">1. You will have only <span>30 seconds</span> per each question.</div>

<div class="info">2. Once you select your answer, it can't be undone.</div>

<div class="info">3. You can't select any option once time goes off.</div>

<div class="info">4. You can't exit from the Quiz while you're playing.</div>

<div class="info">5. You'll get 1 point for each correct answer.</div>

</div>

<div class="buttons">

<button class="quit">Quit</button>

<button class="restart">Next</button>

</div>

</div>

<!-- Quiz Box -->

<div class="quiz\_box">

<header>

<div class="title">INSTANT QUIZ CONTEST</div>

<div class="timer">

<div class="time\_left\_txt">Time Left</div>

<div class="timer\_sec">30</div>

</div>

<div class="time\_line"></div>

</header>

<section>

<div class="que\_text">

<!-- Here I've inserted question from JavaScript -->

</div>

<div class="option\_list">

<!-- Here I've inserted options from JavaScript -->

</div>

</section>

<!-- footer of Quiz Box -->

<footer>

<div class="total\_que">

<!-- Here I've inserted Question Count Number from JavaScript -->

</div>

<button class="next\_btn">Next Question</button>

</footer>

</div>

<!-- Result Box -->

<div class="result\_box">

<div class="icon">

<img src="CMR.PNG" style="width:160px;height:150px;">

</div>

<div class="complete\_text">You've successfully submitted!</div>

<div class="score\_text">

<!-- Here I've inserted Score Result from JavaScript -->

</div>

<div class="buttons">

<button class="restart">Try again!</button>

<button class="quit">Exit Quiz</button>

</div>

</div>

<!-- Inside this JavaScript file I've inserted Questions and Options only -->

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

<!-- Inside this JavaScript file I've coded all Quiz Codes -->

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

</body>

</html>

</html>

**CSS CODE:**

/\* importing google fonts \*/

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');

\*{

margin-top: 0px;

padding-top: 0px;

box-sizing: border-box;

font-family: 'Poppins', sans-serif;

}

body{

background: url("https://wallpaperboat.com/wp-content/uploads/2019/12/orange-and-white-01.jpg");

overflow: hidden;

}

::selection{

color: #fff;

background: #32a852;

}

.qlogo{

position:fixed;

margin-top: 180px;

padding-top: 0px;

left:37%;

height:85%;

}

.xy{

position:fixed;

margin-top: 2px;

padding-top: 5px;

left:17%;

height:60%;

}

/\*.register1\_btn.activeInfo{

position: fixed;

background-color: #fff;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2),

0 6px 20px 0 rgba(0, 0, 0, 0.19);

}

.register\_btn.activeInfo{

position: fixed;

background-color: #fff;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2),

0 6px 20px 0 rgba(0, 0, 0, 0.19);

}\*/

.register\_btn{

position: fixed;

top: 58%;

left: 50%;

padding:20px 20px;

transform: translate(-50%, -50%);

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2),

0 6px 20px 0 rgba(0, 0, 0, 0.19);

}

.register1\_btn input{

color: #fff;

background-color: #007bff;

font-size: 20px;

font-family: 'Poppins', sans-serif;

border-radius: 5px;

border: 1px solid #007bff;

cursor: pointer;

}

.register\_btn p{

padding-bottom: 10px;

}

.register1\_btn{

padding: 0px 45%;

}

.register1\_btn p{

padding-top: 20px;

}

.register\_name{

padding-top: 15px;

}

.register1\_btn{

height: 40px;

width: 100%;

border-top: : 1px solid lightgrey;

display: flex;

align-items: center;

padding-top: 20px;

padding: 0 140px;

border-radius: 5px 5px 0 0;

font-size: 20px;

font-weight: 600;

text-align: center;

padding-bottom: 10px;

font-size: 20px;

font-weight: 600;

}

.register\_header{

height: 40px;

width: 100%;

border-bottom: 1px solid lightgrey;

display: flex;

align-items: center;

padding: 0 140px;

border-radius: 5px 5px 0 0;

font-size: 20px;

font-weight: 600;

text-align: center;

padding-bottom: 10px;

font-size: 20px;

font-weight: 600;

}

.info\_box,

.result\_box{

position: fixed;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2),

0 6px 20px 0 rgba(0, 0, 0, 0.19);

}

.register\_btn{

font-family: 'Poppins', sans-serif;

}

.quiz\_box{

position: fixed;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2),

0 6px 20px 0 rgba(0, 0, 0, 0.19);

}

.start\_btn{

position: fixed;

top: 69%;

left: 50%;

transform: translate(-50%, -50%);

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2),

0 6px 20px 0 rgba(0, 0, 0, 0.19);

}

.register\_btn.activeInfo,

.info\_box.activeInfo,

.quiz\_box.activeQuiz,

.result\_box.activeResult{

opacity: 1;

z-index: 5;

pointer-events: auto;

transform: translate(-50%, -50%) scale(1);

}

.start\_btn button{

font-size: 25px;

font-weight: 500;

font-family: 'Poppins', sans-serif;

color: #fff;

padding: 12px 40px;

outline: none;

border: none;

border-radius: 3px;

background: #323633;

cursor: pointer;

}

.register\_btn{

width: 400px;

background: #fff;

border-radius: 5px;

transform: translate(-50%, -50%) scale(0.9);

opacity: 0;

pointer-events: none;

transition: all 0.3s ease;

}

.info\_box{

width: 540px;

background: #fff;

border-radius: 5px;

transform: translate(-50%, -50%) scale(0.9);

opacity: 0;

pointer-events: none;

transition: all 0.3s ease;

}

.info\_box .info-title{

height: 60px;

width: 100%;

border-bottom: 1px solid lightgrey;

display: flex;

align-items: center;

padding: 0 30px;

border-radius: 5px 5px 0 0;

font-size: 20px;

font-weight: 600;

}

.info\_box .info-list{

padding: 15px 30px;

}

.info\_box .info-list .info{

margin: 5px 0;

font-size: 17px;

}

.info\_box .info-list .info span{

font-weight: 600;

color: #007bff;

}

.info\_box .buttons{

height: 60px;

display: flex;

align-items: center;

justify-content: flex-end;

padding: 0 30px;

border-top: 1px solid lightgrey;

}

.info\_box .buttons button{

margin: 0 5px;

height: 40px;

width: 100px;

font-size: 16px;

font-weight: 500;

cursor: pointer;

border: none;

outline: none;

border-radius: 5px;

border: 1px solid #007bff;

transition: all 0.3s ease;

}

.quiz\_box{

width: 550px;

background: #fff;

border-radius: 5px;

transform: translate(-50%, -50%) scale(0.9);

opacity: 0;

pointer-events: none;

transition: all 0.3s ease;

}

.quiz\_box header{

position: relative;

z-index: 2;

height: 70px;

padding: 0 30px;

background: #fff;

border-radius: 5px 5px 0 0;

display: flex;

align-items: center;

justify-content: space-between;

box-shadow: 0px 3px 5px 1px rgba(0,0,0,0.1);

}

.quiz\_box header .title{

font-size: 20px;

font-weight: 600;

}

.quiz\_box header .timer{

color: #004085;

background: #cce5ff;

border: 1px solid #b8daff;

height: 45px;

padding: 0 8px;

border-radius: 5px;

display: flex;

align-items: center;

justify-content: space-between;

width: 145px;

}

.quiz\_box header .timer .time\_left\_txt{

font-weight: 400;

font-size: 17px;

user-select: none;

}

.quiz\_box header .timer .timer\_sec{

font-size: 18px;

font-weight: 500;

height: 30px;

width: 45px;

color: #fff;

border-radius: 5px;

line-height: 30px;

text-align: center;

background: #343a40;

border: 1px solid #343a40;

user-select: none;

}

.quiz\_box header .time\_line{

position: absolute;

bottom: 0px;

left: 0px;

height: 3px;

background: #007bff;

}

section{

padding: 25px 30px 20px 30px;

background: #fff;

}

section .que\_text{

font-size: 25px;

font-weight: 600;

}

section .option\_list{

padding: 20px 0px;

display: block;

}

section .option\_list .option{

background: aliceblue;

border: 1px solid #84c5fe;

border-radius: 5px;

padding: 8px 15px;

font-size: 17px;

margin-bottom: 15px;

cursor: pointer;

transition: all 0.3s ease;

display: flex;

align-items: center;

justify-content: space-between;

}

section .option\_list .option:last-child{

margin-bottom: 0px;

}

section .option\_list .option:hover{

color: #004085;

background: #cce5ff;

border: 1px solid #b8daff;

}

section .option\_list .option.correct{

color: #155724;

background: #d4edda;

border: 1px solid #c3e6cb;

}

section .option\_list .option.incorrect{

color: #721c24;

background: #f8d7da;

border: 1px solid #f5c6cb;

}

section .option\_list .option.disabled{

pointer-events: none;

}

section .option\_list .option .icon{

height: 26px;

width: 26px;

border: 2px solid transparent;

border-radius: 50%;

text-align: center;

font-size: 13px;

pointer-events: none;

transition: all 0.3s ease;

line-height: 24px;

}

.option\_list .option .icon.tick{

color: #23903c;

border-color: #23903c;

background: #d4edda;

}

.option\_list .option .icon.cross{

color: #a42834;

background: #f8d7da;

border-color: #a42834;

}

footer{

height: 60px;

padding: 0 30px;

display: flex;

align-items: center;

justify-content: space-between;

border-top: 1px solid lightgrey;

}

footer .total\_que span{

display: flex;

user-select: none;

}

footer .total\_que span p{

font-weight: 500;

padding: 0 5px;

}

footer .total\_que span p:first-child{

padding-left: 0px;

}

footer button{

height: 40px;

padding: 0 13px;

font-size: 18px;

font-weight: 400;

cursor: pointer;

border: none;

outline: none;

color: #fff;

border-radius: 5px;

background: #007bff;

border: 1px solid #007bff;

line-height: 10px;

opacity: 0;

pointer-events: none;

transform: scale(0.95);

transition: all 0.3s ease;

}

footer button:hover{

background: #0263ca;

}

footer button.show{

opacity: 1;

pointer-events: auto;

transform: scale(1);

}

.result\_box{

background: #fff;

border-radius: 5px;

display: flex;

padding: 25px 30px;

width: 450px;

align-items: center;

flex-direction: column;

justify-content: center;

transform: translate(-50%, -50%) scale(0.9);

opacity: 0;

pointer-events: none;

transition: all 0.3s ease;

}

.result\_box .icon{

font-size: 100px;

color: #007bff;

margin-bottom: 10px;

}

.result\_box .complete\_text{

font-size: 20px;

font-weight: 500;

}

.result\_box .score\_text span{

display: flex;

margin: 10px 0;

font-size: 18px;

font-weight: 500;

}

.result\_box .score\_text span p{

padding: 0 4px;

font-weight: 600;

}

.result\_box .buttons{

display: flex;

margin: 20px 0;

}

.result\_box .buttons button{

margin: 0 10px;

height: 45px;

padding: 0 20px;

font-size: 18px;

font-weight: 500;

cursor: pointer;

border: none;

outline: none;

border-radius: 5px;

border: 1px solid #007bff;

transition: all 0.3s ease;

}

.buttons button.restart{

color: #fff;

background: #007bff;

}

.buttons button.restart:hover{

background: #0263ca;

}

.buttons button.quit{

color: #007bff;

background: #fff;

}

.buttons button.quit:hover{

color: #fff;

background: #007bff;

}

**JAVA SCRIPT:**

const start\_btn = document.querySelector(".start\_btn button");

const register\_btn = document.querySelector(".register\_btn");

const register1\_btn = register\_btn.querySelector(".register1\_btn");

const info\_box = document.querySelector(".info\_box");

const exit\_btn = info\_box.querySelector(".buttons .quit");

const continue\_btn = info\_box.querySelector(".buttons .restart");

const quiz\_box = document.querySelector(".quiz\_box");

const result\_box = document.querySelector(".result\_box");

const option\_list = document.querySelector(".option\_list");

const time\_line = document.querySelector("header .time\_line");

const timeText = document.querySelector(".timer .time\_left\_txt");

const timeCount = document.querySelector(".timer .timer\_sec");

// if startQuiz button clicked

/\*start\_btn.onclick = ()=>{

register\_btn.classList.add("activeInfo"); //show register box

}

register\_btn.register.onclick = ()=>{

info\_box.classList.add("activeInfo"); //show info box

}\*/

start\_btn.onclick = ()=>{

register\_btn.classList.add("activeInfo"); //show register box

}

/\*register\_btn.onclick=()=>{

info\_box.classList.add("activeInfo");

}\*/

register1\_btn.onclick = ()=>{

register\_btn.classList.remove("activeInfo");

info\_box.classList.add("activeInfo");

}

// if exitQuiz button clicked

exit\_btn.onclick = ()=>{

info\_box.classList.remove("activeInfo"); //hide info box

}

// if continueQuiz button clicked

continue\_btn.onclick = ()=>{

info\_box.classList.remove("activeInfo"); //hide info box

quiz\_box.classList.add("activeQuiz"); //show quiz box

showQuetions(0); //calling showQestions function

queCounter(1); //passing 1 parameter to queCounter

startTimer(30); //calling startTimer function

startTimerLine(0); //calling startTimerLine function

}

let timeValue = 30;

let que\_count = 0;

let que\_numb = 1;

let userScore = 0;

let counter;

let counterLine;

let widthValue = 0;

const restart\_quiz = result\_box.querySelector(".buttons .restart");

const quit\_quiz = result\_box.querySelector(".buttons .quit");

// if restartQuiz button clicked

restart\_quiz.onclick = ()=>{

quiz\_box.classList.add("activeQuiz"); //show quiz box

result\_box.classList.remove("activeResult"); //hide result box

timeValue = 30;

que\_count = 0;

que\_numb = 1;

userScore = 0;

widthValue = 0;

showQuetions(que\_count); //calling showQestions function

queCounter(que\_numb); //passing que\_numb value to queCounter

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

startTimer(timeValue); //calling startTimer function

startTimerLine(widthValue); //calling startTimerLine function

timeText.textContent = "Time Left"; //change the text of timeText to Time Left

next\_btn.classList.remove("show"); //hide the next button

}

// if quitQuiz button clicked

quit\_quiz.onclick = ()=>{

window.location.reload(); //reload the current window

}

const next\_btn = document.querySelector("footer .next\_btn");

const bottom\_ques\_counter = document.querySelector("footer .total\_que");

// if Next Que button clicked

next\_btn.onclick = ()=>{

if(que\_count < questions.length - 1){ //if question count is less than total question length

que\_count++; //increment the que\_count value

que\_numb++; //increment the que\_numb value

showQuetions(que\_count); //calling showQestions function

queCounter(que\_numb); //passing que\_numb value to queCounter

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

startTimer(timeValue); //calling startTimer function

startTimerLine(widthValue); //calling startTimerLine function

timeText.textContent = "Time Left"; //change the timeText to Time Left

next\_btn.classList.remove("show"); //hide the next button

}else{

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

showResult(); //calling showResult function

}

}

// getting questions and options from array

function showQuetions(index){

const que\_text = document.querySelector(".que\_text");

//creating a new span and div tag for question and option and passing the value using array index

let que\_tag = '<span>'+ questions[index].numb + ". " + questions[index].question +'</span>';

let option\_tag = '<div class="option"><span>'+ questions[index].options[0] +'</span></div>'

+ '<div class="option"><span>'+ questions[index].options[1] +'</span></div>'

+ '<div class="option"><span>'+ questions[index].options[2] +'</span></div>'

+ '<div class="option"><span>'+ questions[index].options[3] +'</span></div>';

que\_text.innerHTML = que\_tag; //adding new span tag inside que\_tag

option\_list.innerHTML = option\_tag; //adding new div tag inside option\_tag

const option = option\_list.querySelectorAll(".option");

// set onclick attribute to all available options

for(i=0; i < option.length; i++){

option[i].setAttribute("onclick", "optionSelected(this)");

}

}

// creating the new div tags which for icons

let tickIconTag = '<div class="icon tick"><i class="fas fa-check"></i></div>';

let crossIconTag = '<div class="icon cross"><i class="fas fa-times"></i></div>';

//if user clicked on option

function optionSelected(answer){

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

let userAns = answer.textContent; //getting user selected option

let correcAns = questions[que\_count].answer; //getting correct answer from array

const allOptions = option\_list.children.length; //getting all option items

if(userAns == correcAns){ //if user selected option is equal to array's correct answer

userScore += 1; //upgrading score value with 1

answer.classList.add("correct"); //adding green color to correct selected option

answer.insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to correct selected option

console.log("Correct Answer");

console.log("Your correct answers = " + userScore);

}else{

answer.classList.add("incorrect"); //adding red color to correct selected option

answer.insertAdjacentHTML("beforeend", crossIconTag); //adding cross icon to correct selected option

console.log("Wrong Answer");

for(i=0; i < allOptions; i++){

if(option\_list.children[i].textContent == correcAns){ //if there is an option which is matched to an array answer

option\_list.children[i].setAttribute("class", "option correct"); //adding green color to matched option

option\_list.children[i].insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to matched option

console.log("Auto selected correct answer.");

}

}

}

for(i=0; i < allOptions; i++){

option\_list.children[i].classList.add("disabled"); //once user select an option then disabled all options

}

next\_btn.classList.add("show"); //show the next button if user selected any option

}

function showResult(){

info\_box.classList.remove("activeInfo"); //hide info box

quiz\_box.classList.remove("activeQuiz"); //hide quiz box

result\_box.classList.add("activeResult"); //show result box

const scoreText = result\_box.querySelector(".score\_text");

if (userScore > 3){ // if user scored more than 3

//creating a new span tag and passing the user score number and total question number

let scoreTag = '<span>and congrats! , You got <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';

scoreText.innerHTML = scoreTag; //adding new span tag inside score\_Text

}

else if(userScore > 1){ // if user scored more than 1

let scoreTag = '<span>and nice , You got <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';

scoreText.innerHTML = scoreTag;

}

else{ // if user scored less than 1

let scoreTag = '<span>and sorry , You got only <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';

scoreText.innerHTML = scoreTag;

}

}

function startTimer(time){

counter = setInterval(timer, 1000);

function timer(){

timeCount.textContent = time; //changing the value of timeCount with time value

time--; //decrement the time value

if(time < 9){ //if timer is less than 9

let addZero = timeCount.textContent;

timeCount.textContent = "0" + addZero; //add a 0 before time value

}

if(time < 0){ //if timer is less than 0

clearInterval(counter); //clear counter

timeText.textContent = "Time Off"; //change the time text to time off

const allOptions = option\_list.children.length; //getting all option items

let correcAns = questions[que\_count].answer; //getting correct answer from array

for(i=0; i < allOptions; i++){

if(option\_list.children[i].textContent == correcAns){ //if there is an option which is matched to an array answer

option\_list.children[i].setAttribute("class", "option correct"); //adding green color to matched option

option\_list.children[i].insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to matched option

console.log("Time Off: Auto selected correct answer.");

}

}

for(i=0; i < allOptions; i++){

option\_list.children[i].classList.add("disabled"); //once user select an option then disabled all options

}

next\_btn.classList.add("show"); //show the next button if user selected any option

}

}

}

function startTimerLine(time){

counterLine = setInterval(timer, 56);

function timer(){

time += 1; //upgrading time value with 1

time\_line.style.width = time + "px"; //increasing width of time\_line with px by time value

if(time > 549){ //if time value is greater than 549

clearInterval(counterLine); //clear counterLine

}

}

}

function queCounter(index){

//creating a new span tag and passing the question number and total question

let totalQueCounTag = '<span><p>'+ index +'</p> of <p>'+ questions.length +'</p> Questions</span>';

bottom\_ques\_counter.innerHTML = totalQueCounTag; //adding new span tag inside bottom\_ques\_counter

}

**6. CONCLUSION AND REFERENCE:**

**CONCLUSION:**

To conclude, this is a simple Online MCQ Quiz which will give instant results

For users. Our proposed solution is very efficient and very easy to understand. This will save so much time for the user. This will give more readability to the user as well as administrator. The marks of student will be calculated according to questions they attempt and w

ill be displayed by the system to teachers and student.

**REFERENCE:**

* Patil, Prateek and Karl Moss, 2017, Java Database Programming with JDBC, Coriclis Group Books.
* Sierra, Kathy and Bert Bates, 2003, Head First Java, O’Reilly.
* Sarcar, Vaskaran, 2018, Java Design Patterns, Apress.
* <https://www.w3schools.com/html/default.asp>
* <https://www.w3schools.com/css/default.asp>
* <https://www.w3schools.com/js/default.asp>