

# **INTERDISCIPLINARY PROJECT REPORT**

at

**Sathyabama Institute of Science and Technology  
(Deemed to be University)**

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

by

**SARVASETTY SAI SATHVIKA  
(Reg. No. 40731087)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF COMPUTING  
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY  
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,  
CHENNAI – 600119, TAMILNADU**

**APRIL 2023**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC  
(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119

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



---

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SARVASETTY SAI SATHVIKA (Reg. No. 40731087)** who carried out the project entitled "**ONLINE QUIZ APPLICATION** " under my supervision from February 2023 to April 2023.

Internal Guide

**Ms.K.ISHWARYA, M.Tech., (Ph.D)**

Head of the Department

**Dr. S. Vigneshwari, M.E., Ph.D**

---

Submitted for Viva voce Examination held on 17-04-2023

Internal Examiner

External Examiner

## **DECLARATION**

I **SARVASETTY SAI SATHVIKA (40731087)** hereby declare that the Project Phase-1 Report entitled “**ONLINE QUIZ APPLICATION**” done by me under the guidance of **Ms.K.ISHWARYA, M.Tech.,(Ph.D)**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering with Specialization in Artificial Intelligence**.

**DATE: 17-04-2023**

**PLACE: Chennai**



**SIGNATURE OF THE CANDIDATE**

## **ACKNOWLEDGEMENT**

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

We convey our thanks to **Dr. T. Sasikala M.E., Ph. D, Dean.**, School of Computing, **Dr. S. Vigneshwari M.E.,Ph.D.**, Head of the Department of Computer Science and Engineering for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Ms.K.ISHWARYA, M.Tech.,(Ph.D) Assistant Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our phase-1 project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

## TRAINING CERTIFICATE



# **FUTURO FOCUS**

## **CERTIFICATE**

### **of Completion**

This Certificate is awarded to :

**SARVASETTY SAI SATHVIKA**

( Registration Number 40731087 )

the student of **III Year B.E Computer Science and  
Engineering (Artificial Intelligence),  
Sathyabama Institute of Science and Technology**  
for successfully completing the Professional Training Course  
on **Web Application Development using Django**  
held from 09.02.2023 to 08.04.2023



**J. SARAVANAKUMAR**

Director  
Futuro Focus

Certificate Number: FF/SU/WEB/0423/036



## **ABSTRACT**

The project: Online quiz Application System is a web-based quiz system for accessing students. This is a collection of number of different types of quizzes like technical, games, sports, etc . A user can access/play all of the quiz and can attempt any of the one. There will be limited number of questions and for each correct answer user will get a credit score. It shows when student get 50% grade then it shows Average grade, You can do better. And when the another student score grade is above 80% then it shows Excellent, Keep the good work going on. User can see answers as well as can ask a query related to it. There are many quiz applications available currently on internet. But there are few Which provide better understanding between users and the application like, providing proper answers, user query solving, uploading user questions as well as answer to it, etc .To develop a user friendly quiz application which will contain : Numbers of quiz , Answers to every question, Query solving regarding any question, Uploading of user question and answer, and to improve the knowledge level of users. IT is a system by which students can sit in a quiz which need no pencil and paper. Nowadays, students take quiz manually. Lecturers need spend more time on grading. Other than that, the quiz paper maybe will be missing. Students need to wait for lecturers finish grading to get their result. To develop an application which will contain solution to the above problems. By this application the user will come to know about his/her level and can learn additional knowledge. Also by this application a user can expand his/her knowledge among the world.

## TABLE OF CONTENTS

| CHAPTER  | TITLE                                                                   | PAGE NO   |
|----------|-------------------------------------------------------------------------|-----------|
|          | Abstract                                                                | i         |
|          | List of Figures                                                         | ii        |
|          | List of Abbreviations                                                   | iii       |
| <b>1</b> | <b>INTRODUCTION</b>                                                     | <b>1</b>  |
|          | 1.1 OBJECTIVES OF THE PROJECT                                           | 1         |
|          | 1.2 ADVANTAGES OF THE QUIZ                                              | 2         |
|          | 1.3 DISADVANTAGES OF THE QUIZ                                           | 3         |
| <b>2</b> | <b>AIM AND SCOPE OF THE PROJECT</b>                                     | <b>4</b>  |
|          | 2.1 LIMITATION OF WORK                                                  | 4         |
|          | 2.2 PROJECT PLANNING                                                    | 4         |
|          | 2.3 EXPECTED OUTCOME                                                    | 5         |
|          | 2.4 METHODS USED                                                        | 5         |
| <b>3</b> | <b>EXPERIMENTAL OR MATERIAL, AND<br/>METHODOLOGIES, ALGORITHMS USED</b> | <b>6</b>  |
|          | 3.1 TOOLS USED                                                          | 6         |
|          | 3.2 IDEATION MAP                                                        | 9         |
|          | 3.3 ARCHITECTURE DIAGRAM                                                | 10        |
| <b>4</b> | <b>RESULT AND PERFORMANCE ANALYSIS</b>                                  | <b>11</b> |
|          | 4.1 ACCURACY (MODEL ANALYSIS)                                           | 11        |
| <b>5</b> | <b>SUMMARY AND CONCLUSIONS</b>                                          | <b>12</b> |
|          | REFERENCE                                                               | 14        |
|          | SCREENSHOTS AND OUTPUTS                                                 | 15        |
|          | SOURCE CODE                                                             | 37        |

## LIST OF FIGURES

| <b>Fig No</b> | <b>Fig Title</b> | <b>Page no</b> |
|---------------|------------------|----------------|
| 3.1           | HTML logo        | 7              |
| 3.2           | CSS logo         | 7              |
| 3.3           | Java script logo | 8              |

## LIST OF ABBREVIATIONS

| ABBREVIATION | EXPANSION                           |
|--------------|-------------------------------------|
| OQS          | Online Quiz System                  |
| HTML         | Hyper Text Markup Language          |
| CSS          | Cascading Style Sheets              |
| JPG          | Joint Photographic Group            |
| JAVASCRIPT   | Adds Dynamic Nature to the Web Page |

# **CHAPTER 1**

## **INTRODUCTION**

Currently, almost all universities in our country using the manual procedure to setup quiz for students. The manual procedure means in every quiz, students must attend university to sit in the quiz at a specific time. After that, lecturers will collect quiz paper. Sometimes, there are students does not attend university for a reason and lecturers assumed the students absent for that quiz. This scenario is unfair for student that missed the quiz. So the suitable solution for this problem is by design a system that all student can sit in a quiz from any location. Online Quiz System(OQS), is a web-based quiz system; a system that can be used by lecturers to evaluate students effectively, efficiently and perfectly. The purpose of Online Quiz System is to save lecturer's time since the answers are automatically marked. Online Quiz System is developed by using a rule-based algorithm. Rule-based algorithm used in this system is short answer based on keyword.

## **OBJECTIVES OF THE PROJECT**

The first problem is lecturers need spend more time on grading. This is because lecturer need to check student paper one by one. The second problem is difficult for students to answer quiz from any location. Quiz give an impact in coursework. Sometime some students have their own reason cannot take quiz on time. The objectives of this project have been defined as we can know whether the goals of the system have been achieved. There are the following objectives that determine the success of this system:

- i) To develop a system that enables the student to answer a quiz online.
- ii) To implement a system that enables the student to answer a quiz from any location.
- iii) To test the system whether it can help the lecturers in conducting a quiz.

## **ADVANTAGES OF QUIZ**

### **Engage your audience**

Engage your audience in a unique and fun way and connect them to your brand or learning material. By creating a returning event, you will improve your consumer loyalty.

### **Large number**

Taking online quizzes makes it possible to have a large number of participants. It could be up to a number of 1000 participants (and even more!). It doesn't matter at what kind of location they take the online quiz as long as they are connected to the Internet.

### **Randomizing questions**

It is significant more easy to randomize your question with just one click than to do it all manually. Randomizing questions and even answers of those questions is not a lot a of work to do with online quizzes. Besides the advantage of time saving, it also helps preventing students from cheating.

### **Quiz results/ gain insight in audience**

Another plus of online quizzes is getting immediate results from your participants. Not only does the creator get the results, the participant itself is also possible to get his or her results. This makes it for the creator easy to identify the gaps in knowledge. The participants are able to view on what kind of section they have to improve themselves.

## **DISADVANTAGES OF QUIZ**

1. No face to face interaction between the students and teacher.
2. User cannot add Questions for others.
3. User cannot see profile of other user.
4. Questions may take time to arrive due to slow internet connection

## **CHAPTER 2**

### **AIM AND SCOPE OF THE PROJECT**

The scope of the system basically means everything that will be covered in the project and who involves in it. It defines clearly the extent of content that will be covered by the whole system. The scope of the study has to be defined at a preliminary stage and that is very important.

- The Scope of this project is very broad in terms of gaining knowledge and sharing knowledge among world.

Few points are:-

- i) Can be used anywhere any time as it is a web based application.
- ii) This application will be used in educational institutions as well as in corporate world.

### **LIMITATION OF WORK**

The system will focus on quiz such as multiple choice question, true false and short answer only. The system can be used by the UNISZA's lecturer and student only. This system is developing on web based so that it only can be open using a web browser not in android or IOS application.

### **PROJECT PLANNING**

Project Schedule is important document to be considered before the project start. It consists of the start and finish of a project. It as a guideline to completed and finished project on time has given. This Online Quiz System takes four months to develop after passing through several major phases in the development of as system.

## **EXPECTED OUTCOME**

This system is expected to be implemented in web-based. In addition, this system more secure because of no chances of leakage of question paper as it only depend on the administrator. Additionally, this system also less time consuming and is more efficient. This is because analysis will be very easy in proposed system as it is automated. Result will be very precise and accurate and will be declared in very short span of time because calculation and evaluations are done by the simulator itself.

## **METHODS USED**

1. A online Web Application, here admin is able to add any number of questions and we used time limit for each question that adds extra interactivity to the webpage that is done by java script function.
2. After finishing the quiz we get the score displayed on the page. we can add any numbers of questions as per our requirement.
3. CSS is used to add colors and style the content and adding background image.

## CHAPTER 3

### EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED

#### TOOLS USED

**HTML :**It is used for creating web pages and web applications. Let us see what is meant by Hypertext Markup Language, and Web page.

**Hyper Text:** Hypertext simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. Hypertext is a way to link two or more web pages (HTML documents) with each other.

**Markup language:** A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

**Web Page:** A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. With the help of HTML only, we can create static web pages.

#### HTML

HTML determines the structure of web pages. This structure alone is not enough to make a web page look good and interactive. So you'll use assisted technologies such as CSS and JavaScript to make your HTML beautiful and add interactivity, respectively.



**Fig 3.1 HTML logo**

## **CSS**

It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.



**Fig 3.2 CSS logo**

## JAVASCRIPT

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform. It mainly adds dynamic nature to our website.



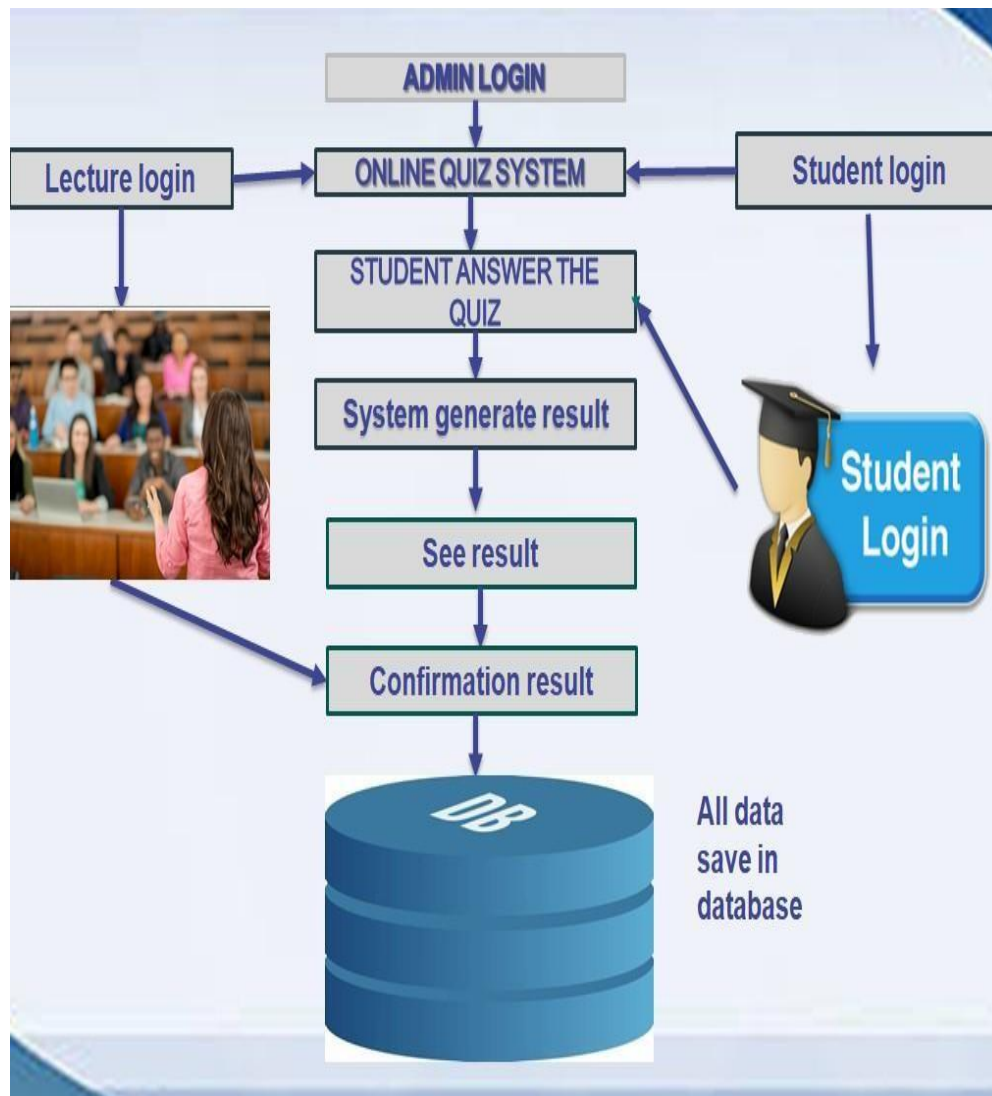
**Fig 3.3 JavaScript logo**

### **What is Rule-Based System?**

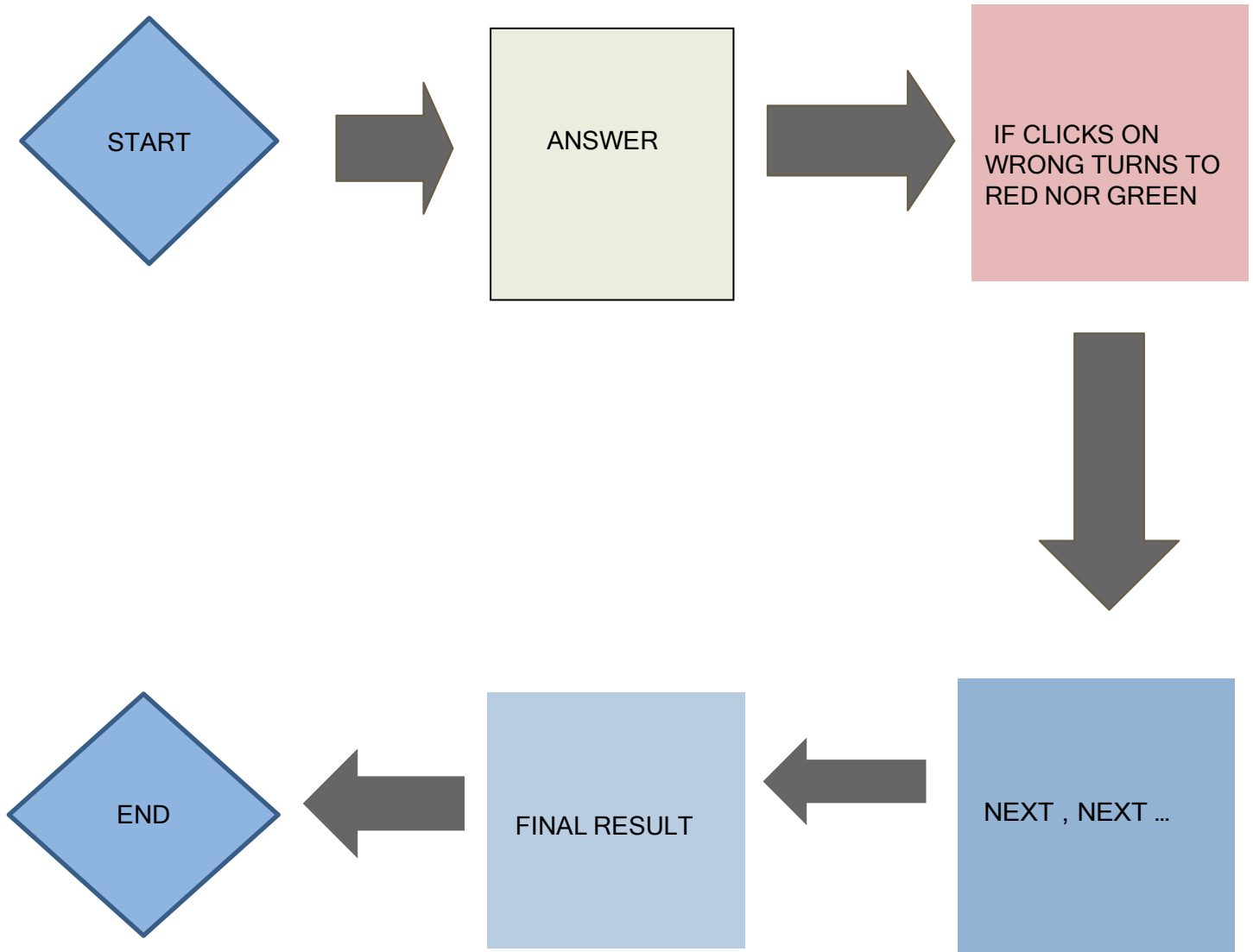
Rule-based systems or production-rules systems are computer systems that use rules to provide recommendations or diagnosis, or to solve a particular problem. A rule-based system consists of the following components:

- i) A database of rules. (Knowledge base)
- ii) A database of facts. (Inputs to the system)
- iii) An interpreter, or inference engine

## IDEATION MAP



## ARCHITECTURE DIAGRAM



## **CHAPTER 4**

### **RESULT AND PERFORMANCE ANALYSIS**

Quiz application provides facility to attempt quiz anywhere and anytime. It save time since user does need to wait for result. So student/user cannot wait for the result. All Student/ user get extra knowledge and skills. Administrator has a privilege to put as much as question in given Quiz in Application. There is instruction page for students to get information about quiz like number of question, given time etc. After that student get the result of quiz. So, student cannot wait for result. also admin is responsible to add, delete and update question in the system. also he can view result of all students. This project more focus on the problems who faced by the lecturers due to the manual system which are available now is not efficient. For instance, the losing of quiz paper, no pencil and no pen to do quiz on time. The lecturer cannot setup quiz for specific time and have to do quiz in class. Lecturer also need spend more time on grading. While, student also cannot take quiz from any location.

#### **4.1 ACCURACY (MODEL ANALYSIS)**

This problems can be solved by Online Quiz System. In addition, this kind of system can be replaced the manual system in order to increase the efficiency of the quiz management. The process in build up this system takes a long time. So that, the insufficient time problems should be faced and solved. In order to produce a good system, the time management is important to ensure the good quality of the system. The improvement should be done and insufficient time should be solved in order to make sure this system can run well and meet the user requirements. It has been a great pleasure for me to work on this exciting and challenging project.

## **CHAPTER 5**

### **SUMMARY AND CONCLUSION**

It discuss project methodology and requirement of software and hardware that guide the system development, it deals with project design and modeling which the core part in the development process. The data flow diagram and the context diagram for this system is shown. Entity relationship diagram is also included to provide better understanding on database design. Fourth chapter will explain the function and flow of the system with interfaces provided, and a few tests are done. In the last chapter which is conclusion, the result has been discussed, concluded and summarized. This project more focus on the problems who faced by the lecturers due to the manual system which are available now is not efficient. For instance, the losing of quiz paper, no pencil and no pen to do quiz on time. The lecturer cannot setup quiz for specific time and have to do quiz in class. Lecture ralso need spend more time on grading. While, student also cannot take quiz from any location. This problems can be solved by Online Quiz System. In addition, this kind of system can be replaced the manual system in order to increase the efficiency of the quiz management.

### **SUMMARY**

This chapter basically deliver the early stages about this project development. It explains more about the initial project development process. This discusses literature review that had been reviewed during feasibility studies. The literature review helps developer to discover the problem of previous research or system which needs to be improves and overcome in this system development. Furthermore, it also helps to gain understanding about the system that undergo the development process.

As a conclusion, Rule-Based algorithm is the most suitable method to use in developing the system. Data Flow Diagram, Entity Relationship Diagram and Data Dictionary are briefly explain for each process that involves in Online Quiz System.

In this chapter, the interfaces of the system that help the user to interact with the system. Other than JavaScript language for web-based system is used to develop this system. Besides that, the testing approach is the methods that are used for test the functional and non-functional of the system for each main module on the system by using test case

In addition, this kind of system can be replaced the manual system in order to increase the efficiency of the quiz management. The process in build up this system takes a long time. So that, the insufficient time problems should be faced and solved. In order to produce a good system, the time management is important to ensure the good quality of the system. The improvement should be done and insufficient time should be solved in order to make sure this system can run well and meet the user requirements.

## REFERENCES

### BOOKS

[1] Rule-based Expert Systems. 22 February 2018.

[2] Umed H. Suthar, Prof. Abdul Rais , Ashish Upadhyay and Prabhakar Upadhya. Online Examination Management System Using Genetic Algorithm. International Journal of Computer Science Trends and Technology (IJCST) – Volume 3 Issue 5, Sep-Oct 2015

[3] Sk. Imran Hossain Shoyeb. Android based Online Quiz Application. Journal from Department of Computer Science and Engineering, East West University, Dhaka-1212, Bangladesh, May 2015.

[4] Hongmei Nie. Design and Development of the Online Examination System Based on B/S Structure. 2nd International Conference on Teaching and Computational Science (ICTCS 2014)

[5] Burcu Yılmaz Kaya , Gökhan Kaya and Metin Dağdeviren. A Sample Application of Web Based Examination System for Distance and Formal Education. Journal of Procedia - Social and Behavioral Sciences 141 ( 2014 ) 1357 – 1362 .

### LINKS

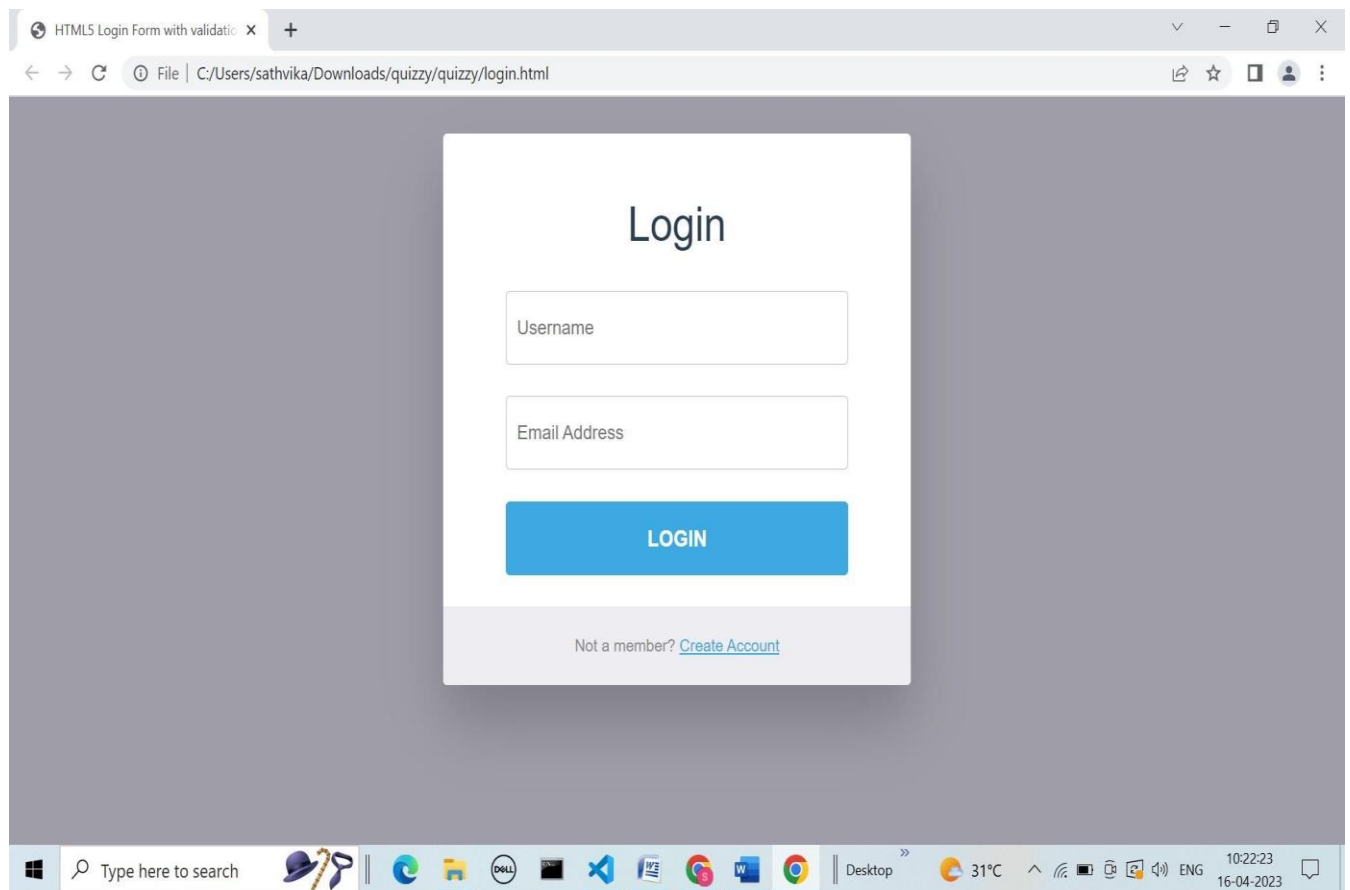
[1] <https://developer.mozilla.org/en-US/docs/Web/HTML>

[2] <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

[3] <https://developer.mozilla.org/en-US/docs/Web/C>

## SCREENSHOTS

### LOGIN PAGE



# HTML

```
index.html - quizz - Visual Studio Code
index.html x # index.css # index.js
index.html > <html> <head> <style> <body> </html>
31 <h1 id="head"> <u> Quizz</u> </h1>
32 
33
34 <div class="bg"></div>
35
36
37
38
39
40 <article class="clock" id="model1">
41 <h3></h3>
42
43 <div class="count">
44 <div id="timer"></div>
45 </div>
46 </article>
47
48
49 <main>
50 <!-- creating a modal for when quiz ends -->
51 <div class="modal-container" id="score-modal">
52
53 <div class="modal-content-container">
54
55 <h1>Congratulations, Quiz Completed.</h1>
56
57 <div class="grade-details">
58 <p>Attempts : 30</p>
59 <p>Wrong Answers : <span id="wrong-answers"></span></p>
60 <p>Right Answers : <span id="right-answers"></span></p>
61 <p>Grade : <span id="grade-percentage"></span>%</p>
62 <p><span id="remarks"></span></p>
-->
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
No problems have been detected in the workspace.
Filter (e.g. text, "P/B", "node_modules")
Ln 11, Col 10 Spaces: 4 UTF-8 CRLF HTML Go Live
Type here to search AQI 96.99 21:03 ENG 06-11-2022
```

```
61 <p>Grade : <span id="grade-percentage"></span>%</p>
62 <p><span id="remarks"></span></p>
63 </div>
64
65 <div class="modal-button-container">
66   <button onclick="closeScoreModal()">Continue</button>
67 </div>
68
69 </div>
70
71
72 <div class="game-quiz-container">
73
74   <div class="game-details-container">
75     <h1>Score : <span id="player-score"></span> / 30</h1>
76     <h1>Question : <span id="question-number"></span> / 30</h1>
77   </div>
78
79   <div class="game-question-container">
80     <h1 id="display-question"></h1>
81   </div>
82
83   <div class="game-options-container">
84     <div class="modal-container" id="option-modal">
85
86       <div class="modal-content-container">
87         <h1>Please Pick An Option</h1>
88
89         <div class="modal-button-container">
90           <button onclick="closeOptionModal()">Continue</button>
91         </div>
92       </div>
93     </div>
94   </div>
95 </div>
```

```
61 <p>Grade : <span id="grade-percentage"></span>%</p>
62 <p><span id="remarks"></span></p>
63 </div>
64
65 <div class="modal-button-container">
66   <button onclick="closeScoreModal()">Continue</button>
67 </div>
68
69 </div>
70
71
72 <div class="game-quiz-container">
73
74   <div class="game-details-container">
75     <h1>Score : <span id="player-score"></span> / 30</h1>
76     <h1>Question : <span id="question-number"></span> / 30</h1>
77   </div>
78
79   <div class="game-question-container">
80     <h1 id="display-question"></h1>
81   </div>
82
83   <div class="game-options-container">
84     <div class="modal-container" id="option-modal">
85
86       <div class="modal-content-container">
87         <h1>Please Pick An Option</h1>
88
89         <div class="modal-button-container">
90           <button onclick="closeOptionModal()">Continue</button>
91         </div>
92       </div>
93     </div>
94   </div>
95 </div>
```

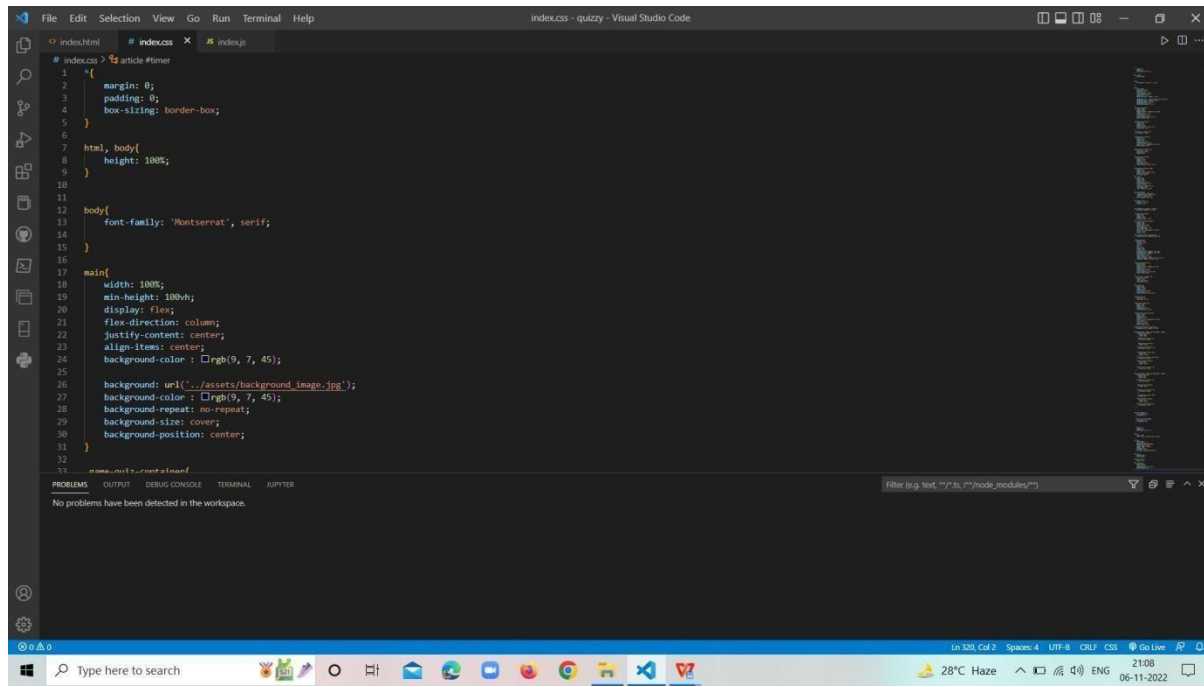
The screenshot shows the Visual Studio Code editor with the file 'index.html' open. The code defines a 'closeOptionModal()' function and a 'Continue' button. Below the button, there are four radio button options, each with a label and a form attribute. The options are 'optionA', 'optionB', 'optionC', and 'optionD'. The 'Continue' button has an 'onclick' event that calls 'closeOptionModal()' and then 'Continue()'. The radio buttons are grouped by the 'name' attribute 'option'.

```
91      <button onclick="closeOptionModal()">Continue</button>
92    </div>
93  </div>
94
95
96 </div>
97
98 <span>
99   <input type="radio" id="option-one" name="option" class="radio" value="optionA" />
100   <label for="option-one" class="option" id="option-one-label"></label>
101 </span>
102
103
104 <span>
105   <input type="radio" id="option-two" name="option" class="radio" value="optionB" />
106   <label for="option-two" class="option" id="option-two-label"></label>
107 </span>
108
109
110 <span>
111   <input type="radio" id="option-three" name="option" class="radio" value="optionC" />
112   <label for="option-three" class="option" id="option-three-label"></label>
113 </span>
114
115
116 <span>
117   <input type="radio" id="option-four" name="option" class="radio" value="optionD" />
118   <label for="option-four" class="option" id="option-four-label"></label>
119 </span>
120
121
122 </div>
```

The screenshot shows the Visual Studio Code editor with the file 'index.html' open. The code defines a 'handleNextQuestion()' function and a 'Next Question' button. Below the button, there is a script tag that calls 'handleNextQuestion()' and then 'Next Question()'. The radio buttons are grouped by the 'name' attribute 'option'.

```
104 <span>
105   <input type="radio" id="option-two" name="option" class="radio" value="optionB" />
106   <label for="option-two" class="option" id="option-two-label"></label>
107 </span>
108
109
110 <span>
111   <input type="radio" id="option-three" name="option" class="radio" value="optionC" />
112   <label for="option-three" class="option" id="option-three-label"></label>
113 </span>
114
115
116 <span>
117   <input type="radio" id="option-four" name="option" class="radio" value="optionD" />
118   <label for="option-four" class="option" id="option-four-label"></label>
119 </span>
120
121
122 </div>
123
124 <div class="next-button-container">
125   <button onclick="handleNextQuestion()">Next Question</button>
126 </div>
127
128 </div>
129
130 </main>
131 <script src="/index.js"></script>
132 </body>
```

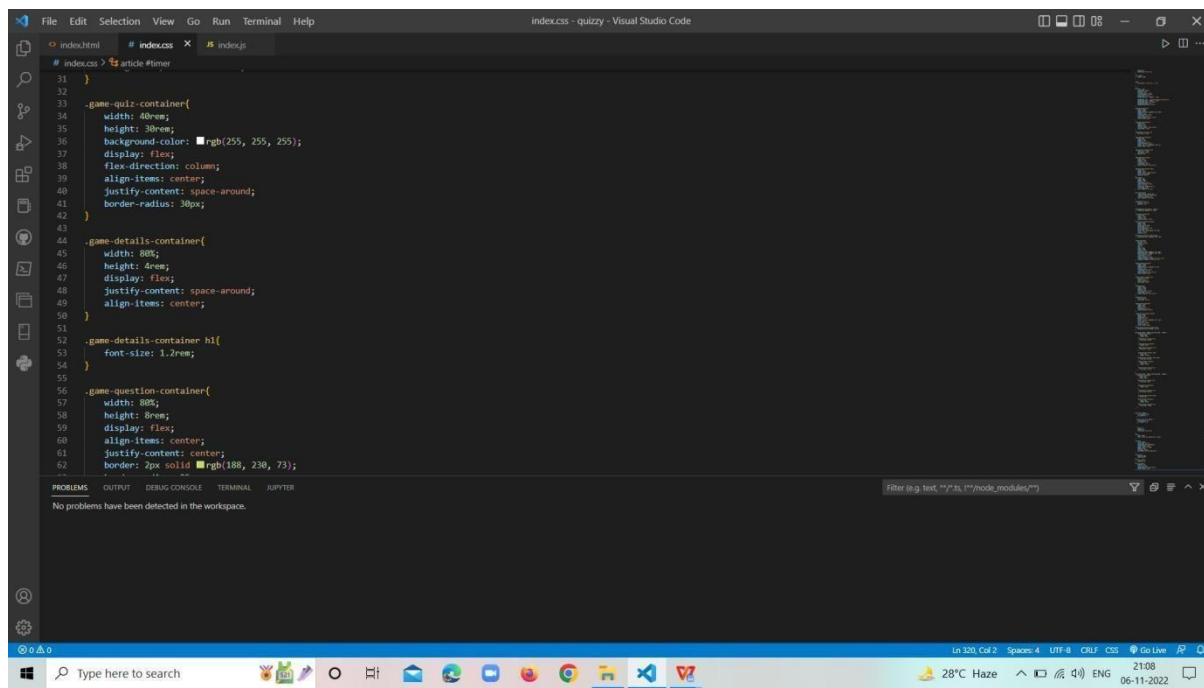
## CSS:



The screenshot shows the Visual Studio Code editor with a file named `index.css` open. The code defines the following CSS rules:

```
1 {  
2   margin: 0;  
3   padding: 0;  
4   box-sizing: border-box;  
5 }  
6  
7 html, body {  
8   height: 100%;  
9 }  
10  
11  
12 body {  
13   font-family: 'Montserrat', serif;  
14 }  
15  
16  
17 main {  
18   width: 100%;  
19   min-height: 100vh;  
20   display: flex;  
21   flex-direction: column;  
22   justify-content: center;  
23   align-items: center;  
24   background-color: #f0f0f0;  
25 }  
26  
27 background: url('../assets/background_image.jpg');  
28 background-color: #f0f0f0;  
29 background-repeat: no-repeat;  
30 background-size: cover;  
31 background-position: center;  
32 }
```

The interface includes a sidebar on the left with icons for Explorer, Search, and Run and Debug. The bottom status bar shows the file is at Line 320, Column 2, with 4 spaces, UTF-8 encoding, and CR LF line endings. The system tray at the bottom indicates a temperature of 28°C, Haze weather, and the date 06-11-2022.



The screenshot shows the Visual Studio Code editor with the same `index.css` file, now displaying additional CSS rules for specific containers:

```
31  
32 }  
33  
34 .game-quiz-container {  
35   width: 40rem;  
36   height: 30rem;  
37   background-color: #f0f0f0;  
38   display: flex;  
39   flex-direction: column;  
40   align-items: center;  
41   justify-content: space-around;  
42   border-radius: 30px;  
43 }  
44  
45 .game-details-container {  
46   width: 80%;  
47   height: 4rem;  
48   display: flex;  
49   justify-content: space-around;  
50   align-items: center;  
51 }  
52  
53 .game-details-container h1 {  
54   font-size: 1.2rem;  
55 }  
56  
57 .game-question-container {  
58   width: 80%;  
59   height: 8rem;  
60   display: flex;  
61   align-items: center;  
62   justify-content: center;  
63   border: 2px solid #f0f0f0;  
64 }
```

The interface is consistent with the previous screenshot, showing the same sidebar and status bar information.

```
# index.css > article #timer
61 justify-content: center;
62 border: 2px solid #188, 230, 73;
63 border-radius: 25px;
64 }
65
66 .game-question-container h1{
67 font-size: 1.1rem;
68 text-align: center;
69 padding: 3px;
70 }
71
72 .game-options-container{
73 width: 80%;
74 height: 12rem;
75 display: flex;
76 flex-wrap: wrap;
77 align-items: center;
78 justify-content: space-around;
79 }
80
81 .game-options-container span{
82 width: 45%;
83 height: 3rem;
84 border: 2px solid #bunlywood;
85 border-radius: 20px;
86 overflow: hidden;
87 }
88 span label{
89 width: 100%;
90 height: 100%;
91 display: flex;
92 align-items: center;
93 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

No problems have been detected in the workspace.

Ln 320, Col 2 Spaces: 4 UTF-8 CR LF CSS Go Live

Type here to search

28°C Haze 21:08 06-11-2022

```
# index.css > article #timer
91 display: flex;
92 align-items: center;
93 justify-content: center;
94 cursor: pointer;
95 transition: transform 0.3s;
96 font-weight: 600;
97 color: #22, 22, 22;
98 }
99
100 span label: hover{
101 ms-transform: scale(1.12);
102 webkit-transform: scale(1.12);
103 transform: scale(1.12);
104 color: #79, 224, 118;
105 }
106
107 input[type='radio'] {
108 position: relative;
109 display: none;
110 }
111
112 input[type='radio']:checked ~ .option {
113 background: #84, 70, 233;
114 }
115
116 .next-button-container{
117 width: 50%;
118 height: 3rem;
119 display: flex;
120 justify-content: center;
121 }
122 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

No problems have been detected in the workspace.

Ln 320, Col 2 Spaces: 4 UTF-8 CR LF CSS Go Live

Type here to search

28°C Haze 21:09 06-11-2022

The screenshot shows the Visual Studio Code editor with the file `index.css` open. The editor displays CSS code for a button and a modal container. The button styles are defined in a `.next-button-container button` selector, and the modal container styles are defined in a `.modal-container` selector. The status bar at the bottom indicates the current cursor position is at line 120, column 2.

```
# index.css > article #timer
120 display: flex;
121 justify-content: center;
122 }
123 .next-button-container button{
124 width: 8rem;
125 height: 2rem;
126 border-radius: 10px;
127 background: none;
128 color: #rgb(25, 25, 25);
129 font-weight: 600;
130 border: 2px solid #rgb(23, 40, 195);
131 cursor: pointer;
132 outline: none;
133 }
134 .next-button-container button:hover{
135 background-color: #rgb(185, 205, 140);
136 }
137 }
138 .modal-container{
139 display: none;
140 position: fixed;
141 z-index: 1;
142 left: 0;
143 top: 0;
144 width: 100%;
145 height: 100%;
146 overflow: auto;
147 background-color: #rgb(22, 144, 128);
148 background-color: #rgba(22, 144, 128);
149 flex-direction: column;
150 align-items: center;
151 justify-content: center;
```

The screenshot shows the Visual Studio Code editor with the file `index.css` open. The editor displays CSS code for a modal content container and grade details. The modal content container styles are defined in a `.modal-content-container` selector, and the grade details styles are defined in a `.grade-details` selector. The status bar at the bottom indicates the current cursor position is at line 150, column 2.

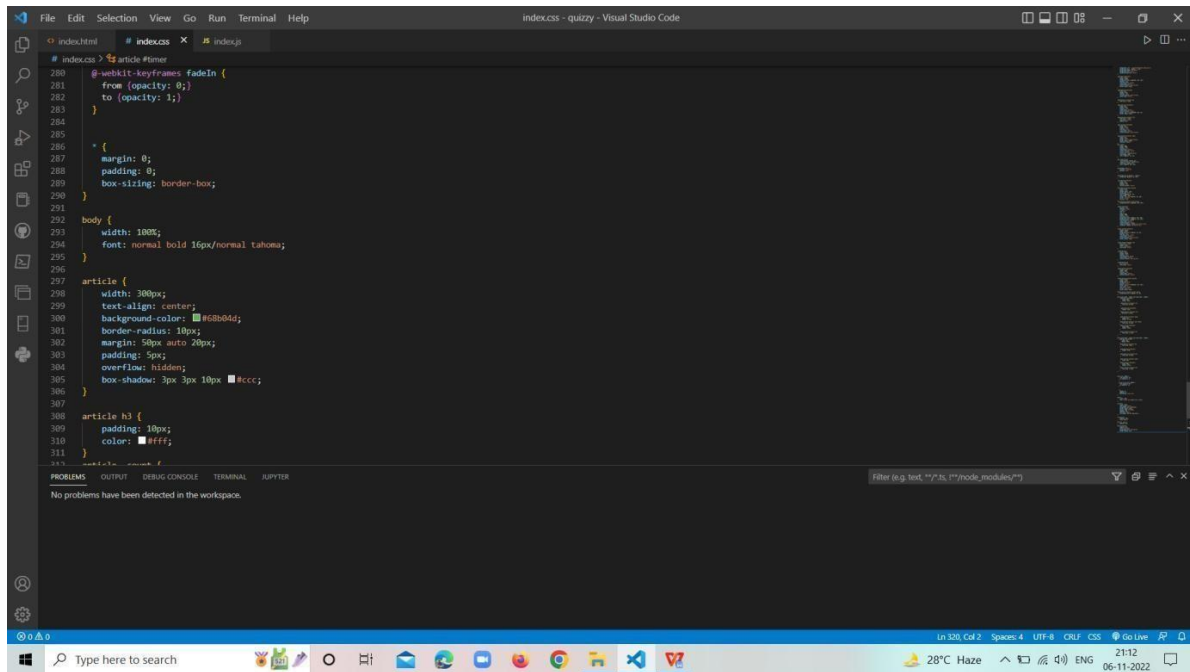
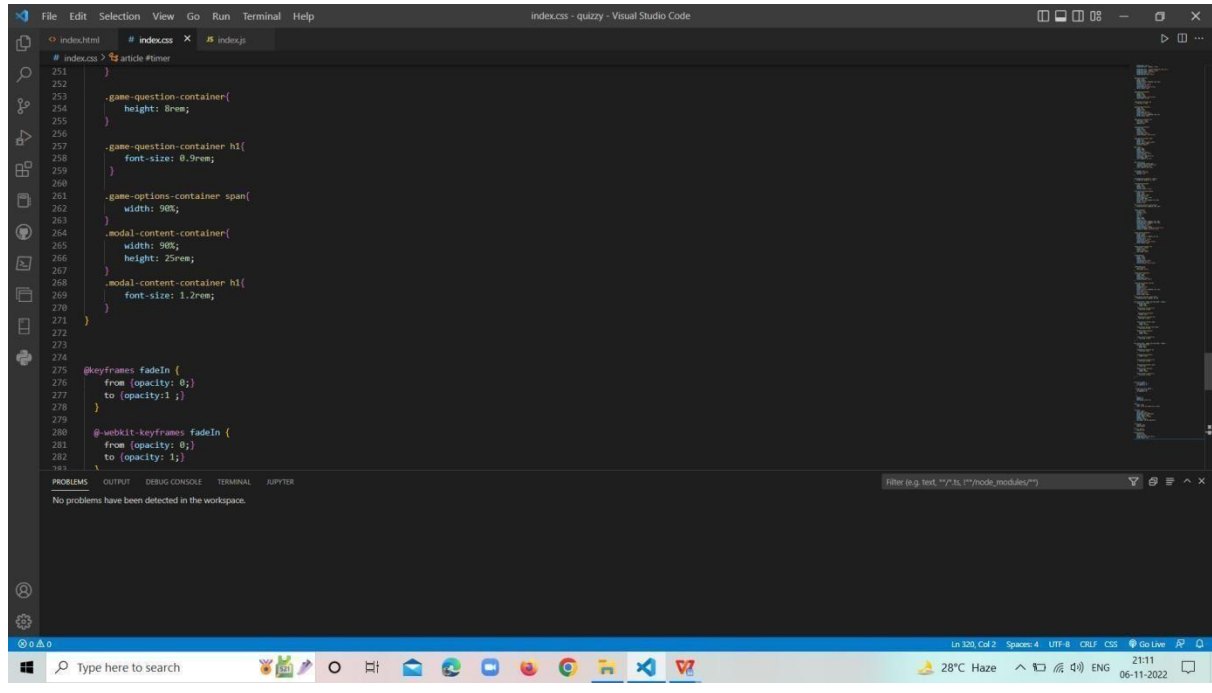
```
# index.css > article #timer
150 align-items: center;
151 justify-content: center;
152 @webkit-animation: fadeIn 1.2s ease-in-out;
153 animation: fadeIn 1.2s ease-in-out;
154 }
155 .modal-content-container{
156 height: 20rem;
157 width: 25rem;
158 background-color: #rgb(43, 42, 42);
159 display: flex;
160 flex-direction: column;
161 align-items: center;
162 justify-content: space-around;
163 border-radius: 25px;
164 }
165 .modal-content-container h1{
166 font-size: 1.3rem;
167 height: 3rem;
168 color: #lightgray;
169 text-align: center;
170 }
171 }
172 }
173 .grade-details{
174 width: 15rem;
175 height: 10rem;
176 display: flex;
177 flex-direction: column;
178 align-items: center;
179 justify-content: space-around;
180 }
181 }
```

The screenshot shows the Visual Studio Code editor with a file named `index.css` open. The code defines styles for a game timer. The `.grade-details` class is a paragraph with white text, centered, with a height of 2rem. The `.modal-button-container` class is a flex container with a height of 2rem, width of 180px, and centered items. It contains a `button` with a height of 18px, width of 2rem, no background, and a border of 1px solid black with a radius of 20px. The button has white text, a font size of 1.1rem, and a cursor pointer. On hover, the background color changes to black. A media query for screens between 380px and 350px wide sets the `.game-quiz-container` to a height of 6rem.

```
181 }
182
183 .grade-details p{
184   color: white;
185   text-align: center;
186 }
187
188 .modal-button-container{
189   height: 2rem;
190   width: 180px;
191   display: flex;
192   align-items: center;
193   justify-content: center;
194 }
195
196 .modal-button-container button{
197   width: 18px;
198   height: 2rem;
199   background: none;
200   outline: none;
201   border: 1px solid black;
202   color: white;
203   font-size: 1.1rem;
204   cursor: pointer;
205   border-radius: 20px;
206 }
207 .modal-button-container button:hover{
208   background-color: black;
209 }
210
211 @media(min-width : 380px) and (max-width : 350px){
212   .game-quiz-container{
213     height: 6rem;
214   }
215 }
```

The screenshot shows the Visual Studio Code editor with a file named `index.css` open. The code defines styles for a game quiz. The `.game-question-container` class is a heading with a font size of 0.9rem. The `.game-options-container` class is a flex container with a width of 90%, height of 2.5rem, and centered items. It contains a `span` with a font size of 0.8rem. The `.modal-content-container` class is a flex container with a width of 90%, height of 25rem, and centered items. It contains a `h1` with a font size of 1.2rem. A media query for screens between 350px and 700px wide sets the `.game-quiz-container` to a width of 90%, height of 80vh, and centered items. It contains a `.game-details-container` with a `h1` of font size 1rem.

```
221   height: 6rem;
222 }
223 .game-question-container h1{
224   font-size: 0.9rem;
225 }
226
227 .game-options-container span{
228   width: 90%;
229   height: 2.5rem;
230 }
231 .game-options-container span label{
232   font-size: 0.8rem;
233 }
234 .modal-content-container{
235   width: 90%;
236   height: 25rem;
237 }
238
239 .modal-content-container h1{
240   font-size: 1.2rem;
241 }
242
243
244 @media(min-width : 350px) and (max-width : 700px){
245   .game-quiz-container{
246     width: 90%;
247     height: 80vh;
248   }
249   .game-details-container h1{
250     font-size: 1rem;
251   }
252 }
```



The screenshot shows the Visual Studio Code editor with a file named `index.css` open. The code defines styles for an article timer. The `article` selector sets a width of 300px, text-align to center, background-color to #f6b04d, border-radius to 10px, margin to 50px auto 20px, padding to 5px, overflow to hidden, and a box-shadow of 3px 3px 10px #ccc. The `article h3` selector sets padding to 10px and color to #fff. The `article .count` selector sets padding to 5px. The `article #timer` selector sets padding to 5px, color to crimson, background-color to #aliceblue, and border-radius to 2px. The bottom status bar shows the file is at line 320, column 2, with 4 spaces, UTF-8 encoding, and CRLF line endings. The system tray at the bottom indicates a temperature of 28°C, haze, and the date 06-11-2022.

```
296
297
298 article {
299   width: 300px;
300   text-align: center;
301   background-color: #f6b04d;
302   border-radius: 10px;
303   margin: 50px auto 20px;
304   padding: 5px;
305   overflow: hidden;
306   box-shadow: 3px 3px 10px #ccc;
307 }
308
309 article h3 {
310   padding: 10px;
311   color: #fff;
312 }
313
314 article .count {
315   padding: 5px;
316 }
317
318 article #timer {
319   padding: 5px;
320   color: crimson;
321   background-color: #aliceblue;
322   border-radius: 2px;
```

## JAVA SCRIPT

The screenshot shows the Visual Studio Code editor with a file named `index.js` open. The code implements a timer function. It selects the `h3` element and sets its innerHTML to "Test ends in:". It sets a variable `sec` to 1800 and a `countDiv` to the element with id "timer". It sets an interval of 1000ms, calling `secpass()` every second. The `secpass()` function uses `use strict`, calculates the remaining seconds (`min` and `sec`), and updates the `countDiv` innerHTML to show the time remaining in minutes and seconds. The status bar shows the file is at line 32, column 33, with 4 spaces, UTF-8 encoding, and CRLF line endings. The system tray at the bottom indicates a temperature of 28°C, cloudy weather, and the date 06-11-2022.

```
1 var h3 = document.getElementsByTagName("h3");
2 h3[0].innerHTML = "Test ends in:";
3
4 var sec = 1800;
5 countDiv = document.getElementById("timer");
6 secpass,
7 countDown = setInterval(function () {
8   "use strict";
9   secpass();
10 }, 1000);
11
12
13 function secpass() {
14   "use strict";
15   var min = Math.floor(sec / 60);
16   var sec = sec % 60;
17   if (sec < 10) {
18     sec = '0' + sec;
19   }
20   if (min < 10) {
21     min = '0' + min;
22   }
23   countDiv.innerHTML = min + ":" + sec;
24   if (sec > 0) {
25     sec = sec - 1;
26   }
27 }
```

```
index.js > ...
10
11
12 if (sec > 0) {
13     sec = sec - 1;
14 }
15
16 } else {
17     clearInterval(countDown);
18     countDiv.innerHTML = 'countdown done';
19 }
20
21 }
22
23 }
24
25 }
26
27 }
28
29 }
30
31 }
32
33 }
34
35 }
36
37 }
38
39 }
40
41 }
42
43 }
44
45 const questions = [
46     {
47         question: "Who developed Python Programming Language?",
48         optionA: "Wick van Rossum",
49         optionB: "Kamus Lendor",
50         optionC: "Guido van Rossum",
51         optionD: "Hiene Stoe",
52         correctOption: "optionC"
53     },
54     {
55         question: "Which type of Programming does Python support?",
56         optionA: "Object-oriented programming",
57         optionB: "structured programming",
58         optionC: "functional programming",
59         optionD: "all of the mentioned",
60         correctOption: "optionD"
61     },
62 ]
```

```
index.js > ...
60     optionD: "all of the mentioned",
61     correctOption: "optionD"
62 },
63
64 {
65     question: "Who was the first President of USA ?",
66     optionA: "Donald Trump",
67     optionB: "Barack Obama",
68     optionC: "Abraham Lincoln",
69     optionD: "George Washington",
70     correctOption: "optionD"
71 },
72
73 {
74     question: "30 days has _____ ?",
75     optionA: "January",
76     optionB: "December",
77     optionC: "June",
78     optionD: "August",
79     correctOption: "optionC"
80 },
81
82 {
83     question: "How many hours can be found in a day ?",
84     optionA: "30 hours",
85     optionB: "38 hours",
86     optionC: "48 hours",
87     optionD: "24 hours",
88     correctOption: "optionD"
89 },
90
91 {
92     question: "Which is the longest river in the world ?",
```

```
91 {
92   question: "Which is the longest river in the world ?",
93   optionA: "River Nile",
94   optionB: "Long River",
95   optionC: "River Niger",
96   optionD: "Lake Chad",
97   correctOption: "optionA"
98 },
99
100 {
101   question: "_____ is the hottest Continent on Earth ?",
102   optionA: "Oceania",
103   optionB: "Antarctica",
104   optionC: "Africa",
105   optionD: "North America",
106   correctOption: "optionC"
107 },
108
109 {
110   question: "Which country is the largest in the world ?",
111   optionA: "Russia",
112   optionB: "Canada",
113   optionC: "Africa",
114   optionD: "Egypt",
115   correctOption: "optionA"
116 },
117
118 {
119   question: "Which of these numbers is an even number ?",
120   optionA: "nine",
121   optionB: "five",
122   optionC: "eight",
123   correctOption: "optionC"
124 },
```

```
121 optionB: "five",
122 optionC: "eight",
123 optionD: "eleven",
124 correctOption: "optionC"
125 },
126
127 {
128   question: "You Can't see me" is a popular saying by",
129   optionA: "Einstein",
130   optionB: "Bill Gates",
131   optionC: "Chris Brown",
132   optionD: "John Cena",
133   correctOption: "optionD"
134 },
135
136 {
137   question: "Which of the following declarations is incorrect?",
138   optionA: "x = 2",
139   optionB: "x = 8",
140   optionC: "xyz = 5",
141   optionD: "None",
142   correctOption: "optionD"
143 },
144
145 {
146   question: "Is Python case sensitive when dealing with identifiers?",
147   optionA: "No",
148   optionB: "Yes",
149   optionC: "machine dependent",
150   optionD: "None",
151   correctOption: "optionA"
152 },
```

```
151 correctOption: "optionA"
152 },
153
154 {
155   question: "Which of the following is the correct extension of the Python file?",
156   optionA: ".python",
157   optionB: ".py",
158   optionC: ".p",
159   optionD: ".pyt",
160   correctOption: "optionB"
161 },
162
163 {
164   question: "Which keyword is used for function in Python language?",
165   optionA: "Function",
166   optionB: "fun",
167   optionC: "Define",
168   optionD: "Def",
169   correctOption: "optionD"
170 },
171
172 {
173   question: "What does pip stand for python?",
174   optionA: "Preferred Installer Program",
175   optionB: "unlimited length",
176   optionC: "all private members must have leading and trailing underscores",
177   optionD: "Program Installer Package",
178   correctOption: "optionA"
179 },
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
180 },
181
182 {
183   question: "In which language is Python written?",
184   optionA: "English",
185   optionB: "Html",
186   optionC: "C",
187   optionD: "R",
188   correctOption: "optionC"
189 },
190
191 {
192   question: "Which of the following functions is a built-in function in python?",
193   optionA: "print()",
194   optionB: "opt()",
195   optionC: "scan()",
196   optionD: "seed()",
197   correctOption: "optionA"
198 },
199
200 {
201   question: "Los Angeles is also known as ?",
202   optionA: "Angels city",
203   optionB: "Shining city",
204   optionC: "City of Angels",
205   optionD: "Lost Angels",
206   correctOption: "optionC"
207 },
208
209 {
210   question: "Which of the following is not a core data type in Python programming?",
211   optionA: "Tuples",
212   optionB: "List",
213   optionC: "Set",
214   optionD: "Dictionary",
215   correctOption: "optionA"
216 },
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
210 question: "Which of the following is not a core data type in Python programming?",
211 optionA: "Tuples",
212 optionB: "Lists",
213 optionC: "Dictionary",
214 optionD: "Class",
215 correctOption: "optionD"
216 },
217
218 {
219   question: "How many sides does an hexagon have ?",
220   optionA: "Six",
221   optionB: "Seveme",
222   optionC: "Four",
223   optionD: "Five",
224   correctOption: "optionA"
225 },
226
227 {
228   question: "How many planets are currently in the solar system ?",
229   optionA: "Eleven",
230   optionB: "Seven",
231   optionC: "Nine",
232   optionD: "Eight",
233   correctOption: "optionD"
234 },
235
236 {
237   question: "Which Planet is the hottest ?",
238   optionA: "Jupitar",
239   optionB: "Mercury",
240   optionC: "Earth",
241   optionD: "Venus",
242 }
```

```
243 },
244
245 {
246   question: "where is the smallest bone in human body located?",
247   optionA: "Toes",
248   optionB: "Ears",
249   optionC: "Fingers",
250   optionD: "Nose",
251   correctOption: "optionB"
252 },
253
254 {
255   question: "How many teeth does an adult human have ?",
256   optionA: "28",
257   optionB: "38",
258   optionC: "32",
259   optionD: "36",
260   correctOption: "optionC"
261 },
262
263 {
264   question: "How many players are allowed in a cricket match ?",
265   optionA: "18 players",
266   optionB: "11 players",
267   optionC: "9 players",
268   optionD: "12 players",
269   correctOption: "optionB"
270 },
271 }
```

```
240     optionC: "Earth",
241     optionD: "Venus",
242     correctOption: "optionB"
243   },
244
245   {
246     question: "Where is the smallest bone in human body located?",
247     optionA: "Toes",
248     optionB: "Ears",
249     optionC: "Fingers",
250     optionD: "Nose",
251     correctOption: "optionB"
252   },
253
254   {
255     question: "How many teeth does an adult human have ?",
256     optionA: "28",
257     optionB: "36",
258     optionC: "32",
259     optionD: "36",
260     correctOption: "optionC"
261   },
262
263   {
264     question: "How many players are allowed in a cricket match ?",
265     optionA: "10 players",
266     optionB: "11 players",
267     optionC: "9 players",
268     optionD: "12 players",
269     correctOption: "optionB"
270   },
271 }
```

```
301 {
302   question: "Which country is the largest in the world ?",
303   optionA: "Russia",
304   optionB: "Canada",
305   optionC: "Africa",
306   optionD: "Egypt",
307   correctOption: "optionA"
308 },
309
310 {
311   question: "Which of these numbers is an odd number ?",
312   optionA: "Ten",
313   optionB: "Twelve",
314   optionC: "Eight",
315   optionD: "Eleven",
316   correctOption: "optionD"
317 },
318
319 {
320   question: "Where is the world tallest building located ?",
321   optionA: "Africa",
322   optionB: "California",
323   optionC: "Dubai",
324   optionD: "Italy",
325   correctOption: "optionC"
326 },
327
328 {
329   question: "What is the maximum possible length of an identifier?",
330   optionA: "15",
331   optionB: "14",
332   correctOption: "optionA"
333 },
334 }
```

The screenshot shows a Visual Studio Code editor window titled 'index.js - quizzy - Visual Studio Code'. The editor is open to a file named 'index.js'. The code is written in JavaScript and includes the following:

```
331     optionA: "15",
332     optionB: "14",
333     optionC: "None of these",
334     optionD: "16",
335     correctOption: "optionC"
336   },
337
338   {
339     question: "In which year was the Python language developed?",
340     optionA: "1991",
341     optionB: "1990",
342     optionC: "1989",
343     optionD: "1992",
344     correctOption: "optionC"
345   },
346
347 ]
348
349
350 let shuffledQuestions = [] //empty array to hold shuffled selected questions
351
352 function handleQuestions() {
353   //function to shuffle and push 10 questions to shuffledQuestions array
354   while (shuffledQuestions.length <= 29) {
355     const random = questions[Math.floor(Math.random() * questions.length)]
356     if (!shuffledQuestions.includes(random)) {
357       shuffledQuestions.push(random)
358     }
359   }
360 }
361
362
```

The bottom of the editor shows the 'PROBLEMS' panel with the message 'No problems have been detected in the workspace.'

The screenshot shows the same Visual Studio Code editor window, now displaying the continuation of the JavaScript code from the previous image. The code includes:

```
361   }
362 }
363
364 let questionNumber = 1
365 let playerScore = 0
366 let wrongAttempt = 0
367 let indexNumber = 0
368
369 // function for displaying next question in the array to dom
370 function NextQuestion(index) {
371   handleQuestions()
372   const currentQuestion = shuffledQuestions[index]
373   document.getElementById("question-number").innerHTML = questionNumber
374   document.getElementById("player-score").innerHTML = playerScore
375   document.getElementById("display-question").innerHTML = currentQuestion.question;
376   document.getElementById("option-one-label").innerHTML = currentQuestion.optionA;
377   document.getElementById("option-two-label").innerHTML = currentQuestion.optionB;
378   document.getElementById("option-three-label").innerHTML = currentQuestion.optionC;
379   document.getElementById("option-four-label").innerHTML = currentQuestion.optionD;
380 }
381
382
383
384 function checkforAnswer() {
385   const currentQuestion = shuffledQuestions[indexNumber] //gets current Question
386   const currentQuestionAnswer = currentQuestion.correctOption //gets current Question's answer
387   const options = document.getElementsByName("option"); //gets all elements in dom with name of 'option' (in this the radio inputs)
388   let correctOption = null
389
390   options.forEach(option => {
391     if (option.value === currentQuestionAnswer) {
392       //get's the correct's radio input with correct answer
393     }
394   })
395 }
396
```

The bottom of the editor shows the 'PROBLEMS' panel with the message 'No problems have been detected in the workspace.'

```
391     if (option.value === currentQuestionAnswer) {
392         //get's correct's radio input with correct answer
393         correctOption = option.labels[0].id
394     }
395 }
396
397 //checking to make sure a radio input has been checked or an option being chosen
398 if (options[0].checked === false && options[1].checked === false && options[2].checked === false && options[3].checked === false) {
399     document.getElementById('option-modal').style.display = "flex"
400 }
401
402 //checking if checked radio button is same as answer
403 options.forEach(option => {
404     if (option.checked === true && option.value === currentQuestionAnswer) {
405         document.getElementById(correctOption).style.backgroundColor = "green"
406         playerScore++
407         indexNumber++
408         //set to delay question number till when next question loads
409         setTimeout(() => {
410             questionNumber++
411         }, 1000)
412     }
413
414     else if (option.checked && option.value !== currentQuestionAnswer) {
415         const wrongLabelId = option.labels[0].id
416         document.getElementById(wrongLabelId).style.backgroundColor = "red"
417         document.getElementById(correctOption).style.backgroundColor = "green"
418         wrongAttempt++
419         indexNumber++
420         //set to delay question number till when next question loads
421         setTimeout(() => {
422             questionNumber++
423         }, 1000)
424     }
425 })
426
427 //called when the next button is called
428 function handleNextQuestion() {
429     checkForAnswer()
430     uncheckRadioButtons()
431     //delay next question displaying for a second
432     setTimeout(() => {
433         if (indexNumber <= 29) {
434             NextQuestion(indexNumber)
435         }
436         else {
437             handleEndGame()
438         }
439         resetOptionBackground()
440     }, 1000);
441 }
442
443 //sets options background back to null after display the right/wrong colors
444 function resetOptionBackground() {
445     const options = document.getElementsByName("option");
446     options.forEach(option => {
447         document.getElementById(option.labels[0].id).style.backgroundColor = ""
448     })
449 }
```

```
421     //set to delay next question number till when next question loads
422     setTimeout(() => {
423         questionNumber++
424     }, 1000)
425 }
426
427 //called when the next button is called
428 function handleNextQuestion() {
429     checkForAnswer()
430     uncheckRadioButtons()
431     //delay next question displaying for a second
432     setTimeout(() => {
433         if (indexNumber <= 29) {
434             NextQuestion(indexNumber)
435         }
436         else {
437             handleEndGame()
438         }
439         resetOptionBackground()
440     }, 1000);
441 }
442
443 //sets options background back to null after display the right/wrong colors
444 function resetOptionBackground() {
445     const options = document.getElementsByName("option");
446     options.forEach(option => {
447         document.getElementById(option.labels[0].id).style.backgroundColor = ""
448     })
449 }
```

```
451 })
452 }
453
454 // unchecking all radio buttons for next question(can be done with map or foreach loop also)
455 function uncheckRadioButtons() {
456   const options = document.getElementsByName("option");
457   for (let i = 0; i < options.length; i++) {
458     options[i].checked = false;
459   }
460 }
461
462 // function for when all questions being answered
463 function handleEndGame() {
464   let remark = null
465   let remarkColor = null
466
467   // condition check for player remark and remark color
468   if (playerScore <= 10) {
469     remark = "Bad Grades, Keep Practicing."
470     remarkColor = "red"
471   }
472   else if (playerScore >= 11 && playerScore < 20) {
473     remark = "Average Grades, You can do better."
474     remarkColor = "orange"
475   }
476   else if (playerScore >= 21) {
477     remark = "Excellent, Keep the good work going."
478     remarkColor = "green"
479   }
480   const playerGrade = (playerScore / 30) * 100
481
482   //data to display to score board
```

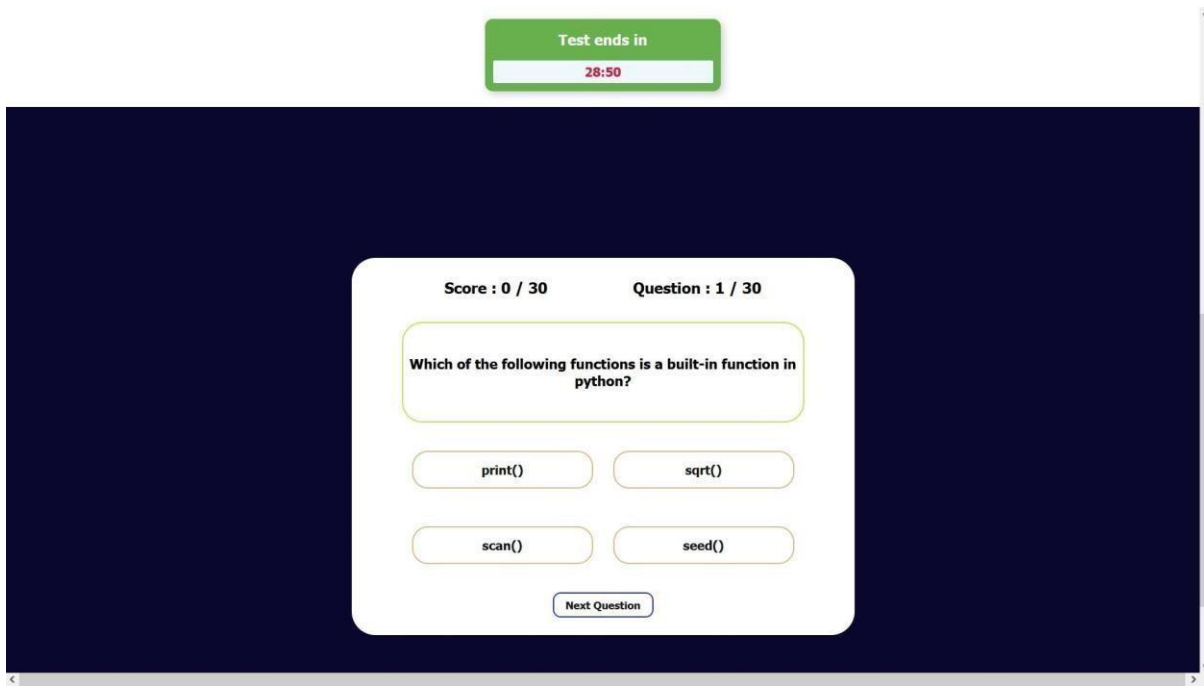
```
476   else if (playerScore >= 21) {
477     remark = "Excellent, Keep the good work going."
478     remarkColor = "green"
479   }
480   const playerGrade = (playerScore / 30) * 100
481
482   //data to display to score board
483   document.getElementById('remarks').innerHTML = remark
484   document.getElementById('remarks').style.color = remarkColor
485   document.getElementById('grade-percentage').innerHTML = playerGrade
486   document.getElementById('wrong-answers').innerHTML = wrongAttempt
487   document.getElementById('right-answers').innerHTML = playerScore
488   document.getElementById('score-modal').style.display = "flex"
489 }
490
491 //closes score modal and resets game
492 function closeScoreModal() {
493   questionNumber = 1
494   playerScore = 0
495   wrongAttempt = 0
496   indexNumber = 0
497   shuffledQuestions = []
498   nextQuestion(indexNumber)
499   document.getElementById('score-modal').style.display = "none"
500 }
501
502 //function to close warning modal
503 function closeOptionModal() {
504   document.getElementById('option-modal').style.display = "none"
505 }
506 }
```

## OUTPUT

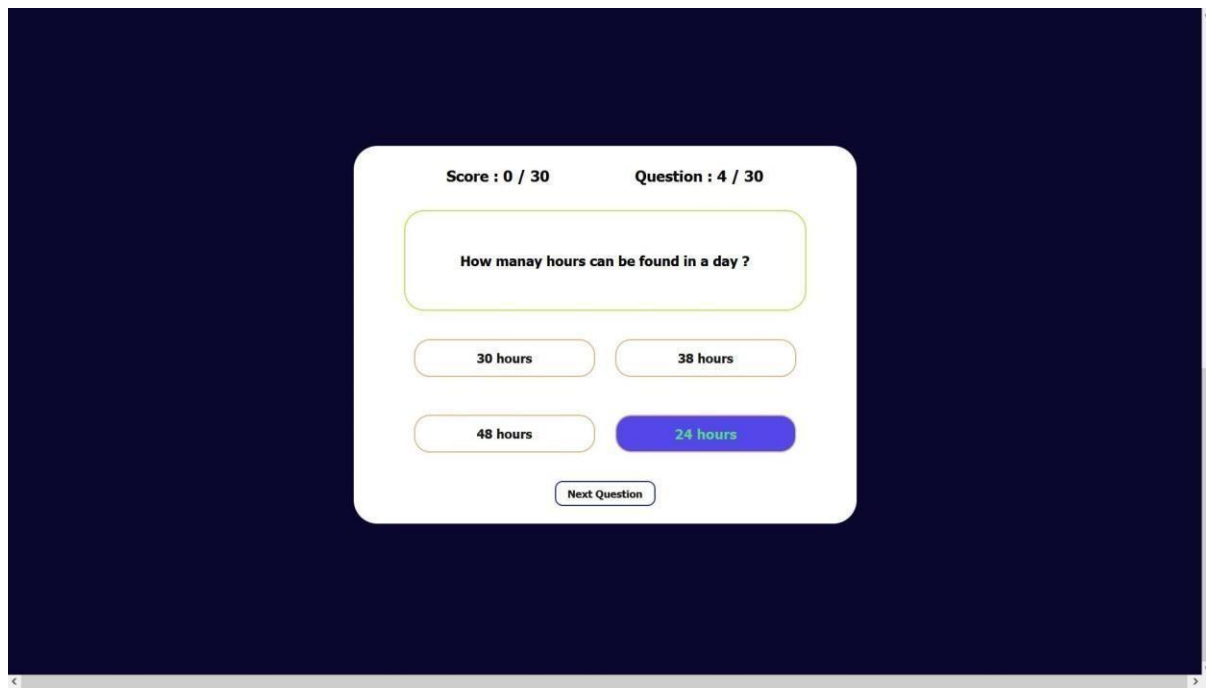
### OPENING PAGE



### TIMER START



**SELECT ONE**



Score : 0 / 30      Question : 4 / 30

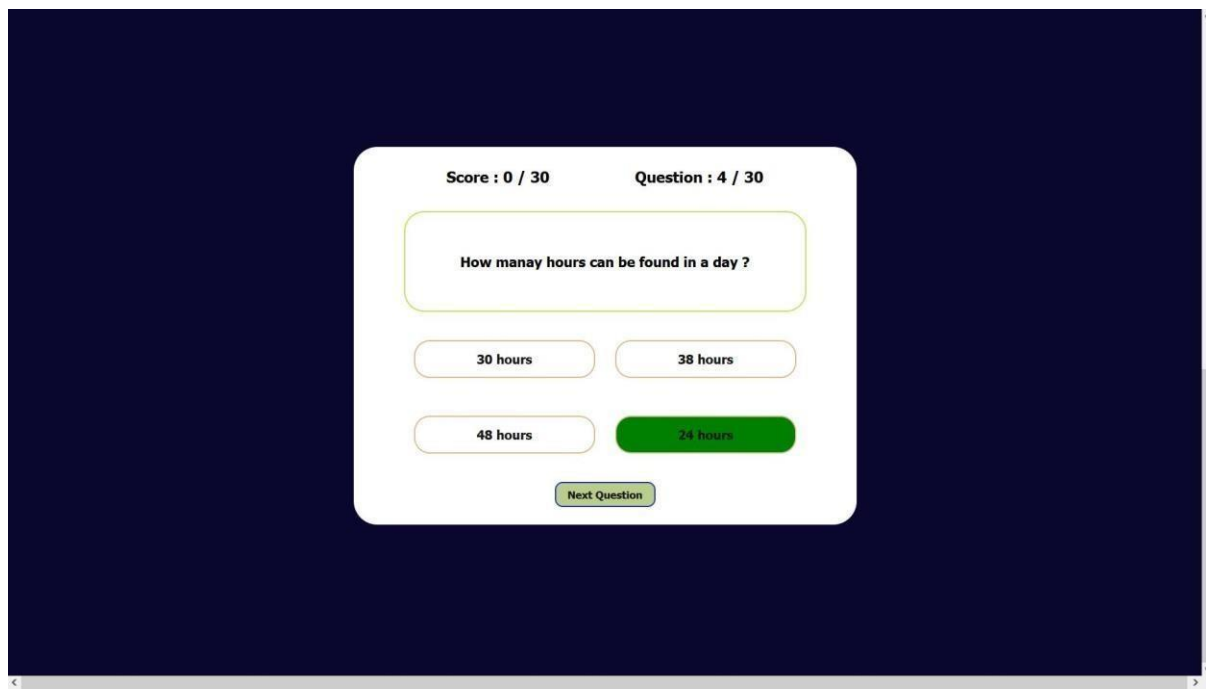
How many hours can be found in a day ?

30 hours      38 hours

48 hours      24 hours

Next Question

**CORRECT ONE**



Score : 0 / 30      Question : 4 / 30

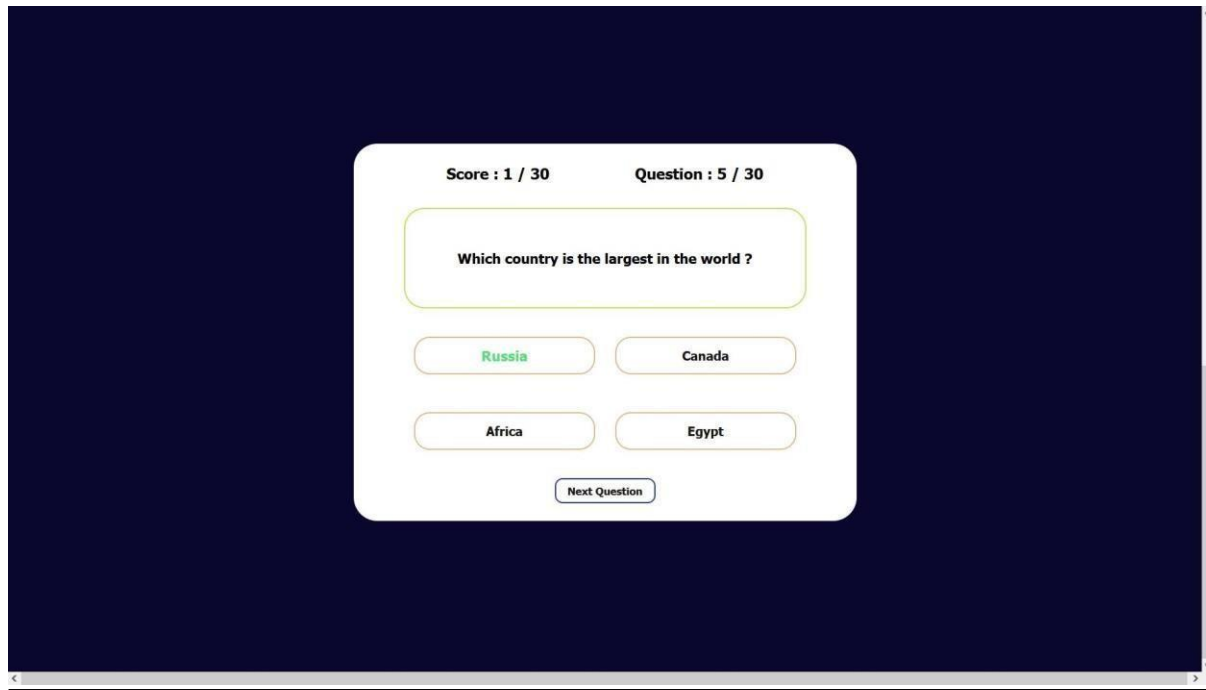
How many hours can be found in a day ?

30 hours      38 hours

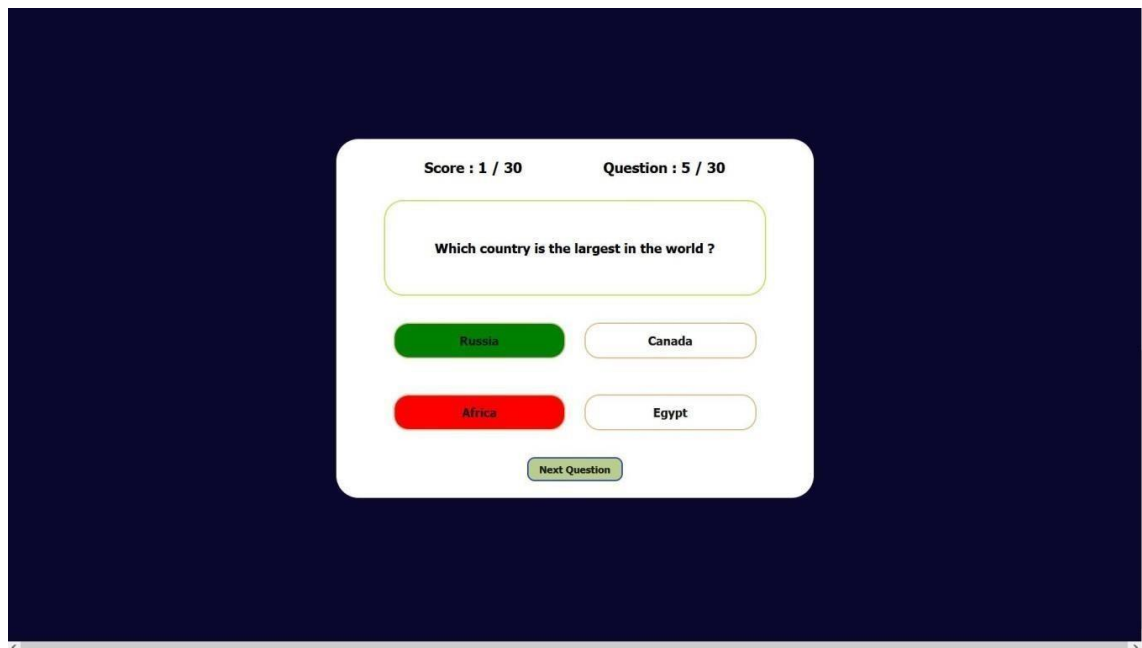
48 hours      24 hours

Next Question

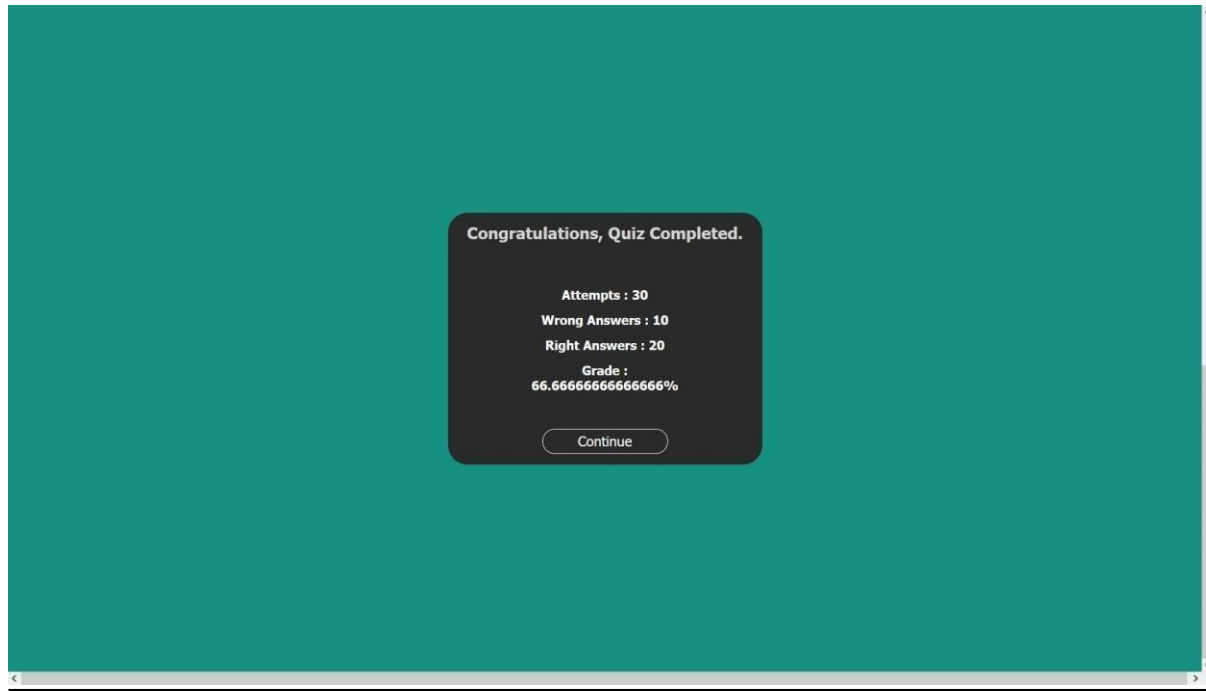
## CURSOR SHOWING ONE



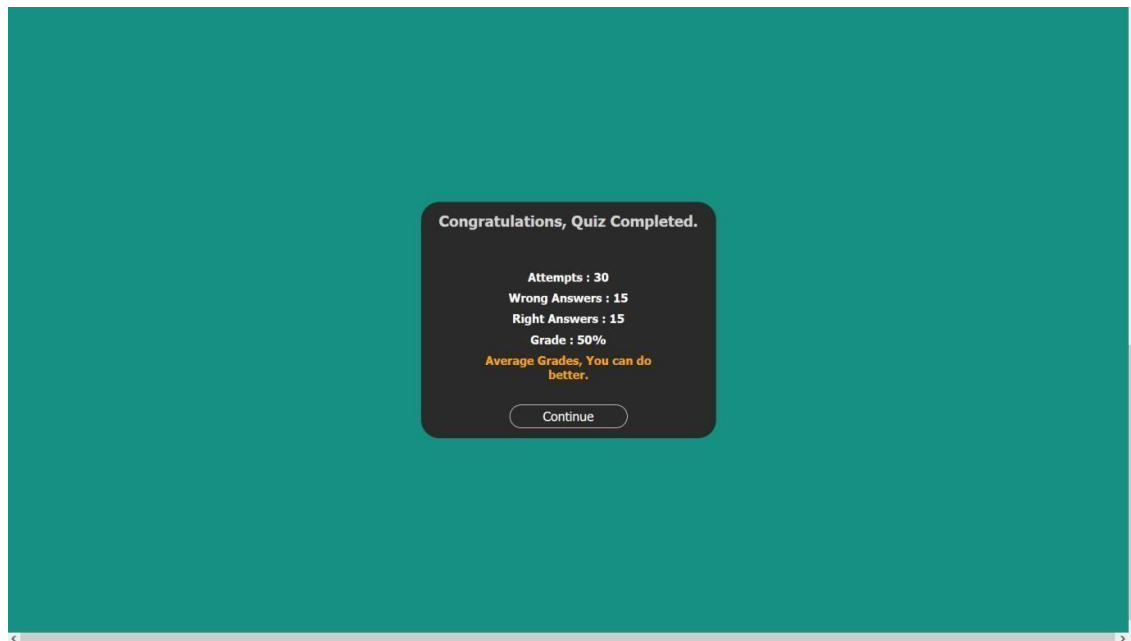
## WRONG ONE



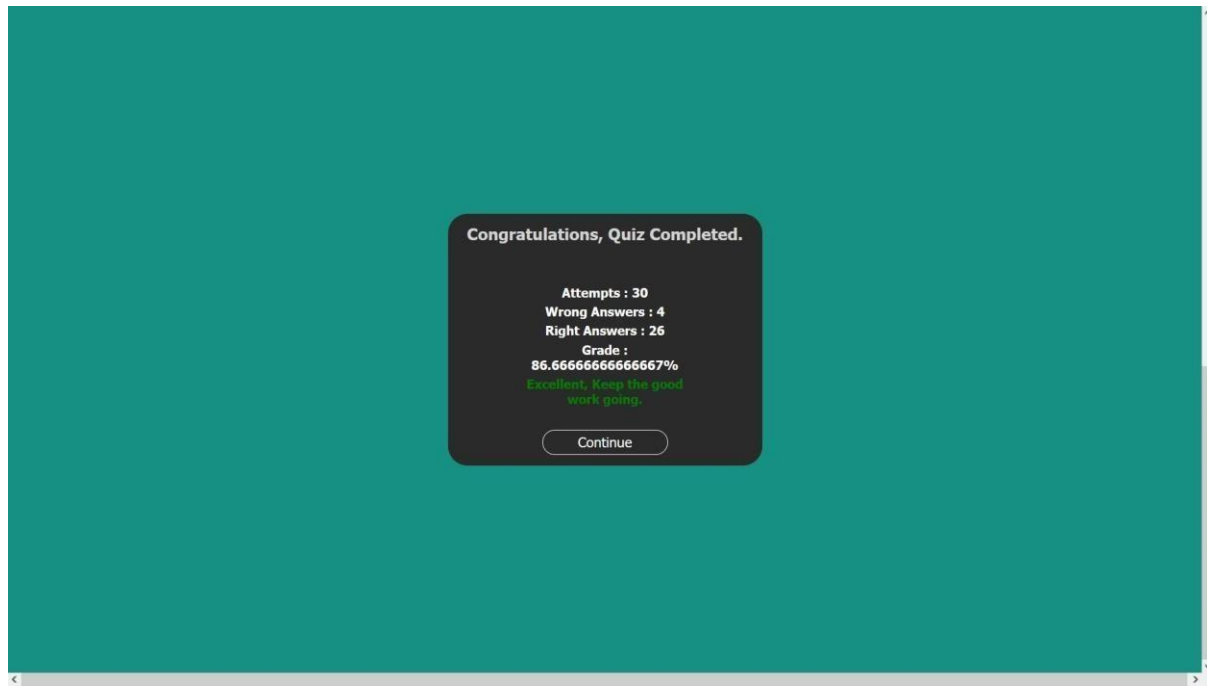
## AVERAGE GRADE



## LOWER GRADE



## HIGHER GRADE



## SOURCE CODE

### HTML CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Quizzy</title>
  <link href="\Users\sathvika\Downloads\quizzy\quizzy\index.css"
rel="stylesheet" type="text/css">
```

```

h1 {text-align: center;}
body, html {
    height: 100%;
    margin: 0;
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
}
#head {

    font-size: 75%;
    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

}

```

```

</style>
</head>
<body onload="NextQuestion(0)">
    

    <div class="bg"></div>

    <article class="clock" id="model3">
        <h3></h3>

        <div class="count">
            <div id="timer"></div>

```

```
</div/>
</article
```

```
<main>
  <!-- creating a modal for when quiz ends -->
  <div class="modal-container" id="score-modal">

    <div class="modal-content-container">

      <h1>Congratulations, Quiz Completed.</h1>

      <div class="grade-details">
        <p>Attempts : 30</p>
        <p>Wrong Answers : <span id="wrong-answers"></span></p>
        <p>Right Answers : <span id="right-answers"></span></p>
        <p>Grade : <span id="grade-percentage"></span>%</p>
        <p><span id="remarks"></span></p>
      </div>

      <div class="modal-button-container">
        <button onclick="closeScoreModal()">Continue</button>
      </div>

    </div>
  </div>

  <div class="game-quiz-container">

    <div class="game-details-container">
```

```

        <h1>Score : <span id="player-score"></span> / 30</h1>
        <h1> Question : <span id="question-number"></span> / 30</h1>
    </div>
<div class="game-question-container">
    <h1 id="display-question"></h1>
</div>

<div class="game-options-container">

    <div class="modal-container" id="option-modal">

        <div class="modal-content-container">
            <h1>Please Pick An Option</h1>

            <div class="modal-button-container">
                <button onclick="closeOptionModal()">Continue</button>
            </div>

        </div>

    </div>

</div>

<span>
    <input type="radio" id="option-one" name="option" class="radio"
value="optionA" />
    <label for="option-one" class="option" id="option-one-label"></label>
</span>

<span>
    <input type="radio" id="option-two" name="option" class="radio"

```

```

value="optionB" />
    <label for="option-two" class="option" id="option-two-label"></label>
</span>

<span>
    <input type="radio" id="option-three" name="option" class="radio"
value="optionC" />
    <label for="option-three" class="option" id="option-three-
label"></label>
</span>

    <span>
        <input type="radio" id="option-four" name="option" class="radio"
value="optionD" />
        <label for="option-four" class="option" id="option-four-
label"></label>
    </span>

</div>

<div class="next-button-container">
    <button onclick = "handleNextQuestion()">Next Question</button>
</div>

</div>
</main>
'use strict';
'use strict';

```

```
<script src="\Users\sathvika\Downloads\quizzzy\quizzzy\index.js"></script>
</body>
```

## CSS CODE

```
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
```

```
html, body{
  height: 100%;
}
```

```
body{
  font-family: 'Montserrat', serif;
}
```

```
main{
  width: 100%;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  background-color : rgb(9, 7, 45);
  background: url('../assets/background_image.jpg');
  background-color : rgb(9, 7, 45);
}
```

```
background-repeat: no-repeat;  
background-size: cover;  
background-position: center;  
}
```

```
.game-quiz-container{  
  width: 40rem;  
  height: 30rem;  
  background-color: rgb(255, 255, 255);  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: space-around;  
  border-radius: 30px;  
}
```

```
.game-details-container{  
  width: 80%;  
  height: 4rem;  
  display: flex;  
  justify-content: space-around;  
  align-items: center;  
}
```

```
.game-details-container h1{  
  font-size: 1.2rem;  
}
```

```
.game-question-container{  
  width: 80%;
```

```
height: 8rem;
display: flex;
align-items: center;
justify-content: center;
border: 2px solid rgb(188, 230, 73);
border-radius: 25px;
}
```

```
.game-question-container h1{
  font-size: 1.1rem;
  text-align: center;
  padding: 3px;
}
```

```
.game-options-container{
  width: 80%;
  height: 12rem;
  display: flex;
  flex-wrap: wrap;
  align-items: center;
  justify-content: space-around;
}
```

```
.game-options-container span{
  width: 45%;
  height: 3rem;
  border: 2px solid burlywood;
  border-radius: 20px;
  overflow: hidden;
}
```

```
span label{
  width: 100%;
  height: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
  cursor: pointer;
  transition: transform 0.3s;
  font-weight: 600;
  color: rgb(22, 22, 22);
}
```

```
span label:hover{
  -ms-transform: scale(1.12);
  -webkit-transform: scale(1.12);
  transform: scale(1.12);
  color: rgb(79, 224, 118);
}
```

```
input[type="radio"] {
  position: relative;
  display: none;
}
```

```
input[type=radio]:checked ~ .option {
  background: rgb(84, 70, 233);
}
```

```
.next-button-container{
```

```

width: 50%;
height: 3rem;
display: flex;
justify-content: center;
}
.next-button-container button{
width: 8rem;
height: 2rem;
border-radius: 10px;
background: none;
color: rgb(25, 25, 25);
font-weight: 600;
border: 2px solid rgb(23, 40, 195);
cursor: pointer;
outline: none;
}
.next-button-container button:hover{
background-color: rgb(185, 205, 140);
}

.modal-container{
display: none;
position: fixed;
z-index: 1;
left: 0;
top: 0;
width: 100%;
height: 100%;
overflow: auto;
background-color: rgb(22, 144, 128);

```

```
background-color: rgba(22, 144, 128);
flex-direction: column;
align-items: center;
justify-content: center;
-webkit-animation: fadeIn 1.2s ease-in-out;
animation: fadeIn 1.2s ease-in-out;
}
```

```
.modal-content-container{
  height: 20rem;
  width: 25rem;
  background-color: rgb(43, 42, 42);
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: space-around;
  border-radius: 25px;
}
```

```
.modal-content-container h1{
  font-size: 1.3rem;
  height: 3rem;
  color: lightgray;
  text-align: center;
}
```

```
.grade-details{
  width: 15rem;
  height: 10rem;
  display: flex;
```

```
    flex-direction: column;
    align-items: center;
    justify-content: space-around;
}
```

```
.grade-details p{
    color: white;
    text-align: center;
}
```

```
.modal-button-container{
    height: 2rem;
    width: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
}
```

```
.modal-button-container button{
    width: 10rem;
    height: 2rem;
    background: none;
    outline: none;
    border: 1px solid rgb(252, 242, 241);
    color: white;
    font-size: 1.1rem;
    cursor: pointer;
    border-radius: 20px;
}
```

```
.modal-button-container button:hover{
```

```

background-color: rgb(83, 82, 82);
}

@media(min-width : 300px) and (max-width : 350px){
  .game-quiz-container{
    width: 90%;
    height: 80vh;
  }
  .game-details-container h1{
    font-size: 0.8rem;
  }

  .game-question-container{
    height: 6rem;
  }
  .game-question-container h1{
    font-size: 0.9rem;
  }

  .game-options-container span{
    width: 90%;
    height: 2.5rem;
  }
  .game-options-container span label{
    font-size: 0.8rem;
  }
  .modal-content-container{
    width: 90%;
    height: 25rem;
  }
}

```

```
.modal-content-container h1{  
    font-size: 1.2rem;  
}  
}
```

```
@media(min-width : 350px) and (max-width : 700px){  
    .game-quiz-container{  
        width: 90%;  
        height: 80vh;  
    }  
    .game-details-container h1{  
        font-size: 1rem;  
    }  
}
```

```
.game-question-container{  
    height: 8rem;  
}
```

```
.game-question-container h1{  
    font-size: 0.9rem;  
}
```

```
.game-options-container span{  
    width: 90%;  
}  
.modal-content-container{  
    width: 90%;  
    height: 25rem;  
}  
.modal-content-container h1{
```

```
        font-size: 1.2rem;
    }
}
```

```
@keyframes fadeIn {
    from {opacity: 0;}
    to {opacity:1 ;}
}
```

```
@-webkit-keyframes fadeIn {
    from {opacity: 0;}
    to {opacity: 1;}
}
```

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
```

```
body {
    width: 100%;
    font: normal bold 16px/normal tahoma;
}
```

```
article {
    width: 300px;
```

```

text-align: center;
background-color: #68b04d;
border-radius: 10px;
margin: 50px auto 20px;
padding: 5px;
overflow: hidden;
box-shadow: 3px 3px 10px #ccc;
}

```

```

article h3 {
  padding: 10px;
  color: #fff;
}
article .count {
  padding: 5px;
}
article #timer{
  padding: 5px;
  color: crimson;
  background-color: aliceblue;
  border-radius: 2px;
}

```

## JS CODE

```

var h3 = document.getElementsByTagName("h3");
h3[0].innerHTML = "Test ends in ";

var sec      = 1800,
    countDiv = document.getElementById("timer"),
    secpass,
    countdown = setInterval(function () {

```

```

        'use strict';

        secpass();
    }, 1000);

function secpass() {
    'use strict';

    var min    = Math.floor(sec / 60),
        remSec = sec % 60;

    if (remSec < 10) {

        remSec = '0' + remSec;

    }
    if (min < 10) {

        min = '0' + min;

    }
    countDiv.innerHTML = min + ":" + remSec;

    if (sec > 0) {

        sec = sec - 1;

    } else {

        clearInterval(countDown);
    }
}

```

```

        countDiv.innerHTML = 'countdown done';

    }
}

const questions = [
    {
        question: "Who developed Python Programming Language?",
        optionA: "Wick van Rossum",
        optionB: "Rasmus Lerdorf",
        optionC: "Guido van Rossum",
        optionD: "Niene Stom",
        correctOption: "optionC"
    },

    {
        question: "Which type of Programming does Python support?",
        optionA: "object-oriented programming",
        optionB: "structured programming",
        optionC: "functional programming",
        optionD: "all of the mentioned",
        correctOption: "optionD"
    },

    {
        question: "Who was the first President of USA ?",
        optionA: "Donald Trump",
        optionB: "Barack Obama",
        optionC: "Abraham Lincoln",

```

```
optionD: "George Washington",  
  correctOption: "optionD"  
},
```

```
{  
  question: "30 days has_____?",  
  optionA: "January",  
  optionB: "December",  
  optionC: "June",  
  optionD: "August",  
  correctOption: "optionC"  
},
```

```
{  
  question: "How manay hours can be found in a day ?",  
  optionA: "30 hours",  
  optionB: "38 hours",  
  optionC: "48 hours",  
  optionD: "24 hours",  
  correctOption: "optionD"  
},
```

```
{  
  question: "Which is the longest river in the world ?",  
  optionA: "River Nile",  
  optionB: "Long River",  
  optionC: "River Niger",  
  optionD: "Lake Chad",  
  correctOption: "optionA"  
},
```

```
{  
  question: "_____is the hottest Continent on Earth ?",  
  optionA: "Oceania",  
  optionB: "Antarctica",  
  optionC: "Africa",  
  optionD: "North America",  
  correctOption: "optionC"  
},
```

```
{  
  question: "Which country is the largest in the world ?",  
  optionA: "Russia",  
  optionB: "Canada",  
  optionC: "Africa",  
  optionD: "Egypt",  
  correctOption: "optionA"  
},
```

```
{  
  question: "Which of these numbers is an even number ?",  
  optionA: "nine",  
  optionB: "Five",  
  optionC: "Eight",  
  optionD: "Eleven",  
  correctOption: "optionC"  
},
```

```
{  
  question: ``"You Can't see me" is a popular saying by`,  
  optionA: "Eminem",
```

```

    optionB: "Bill Gates",
    optionC: "Chris Brown",
    optionD: "John Cena",
    correctOption: "optionD"
},

{
    question: "Which of the following declarations is incorrect?",
    optionA: "_x = 2",
    optionB: "__x = 8",
    optionC: "_xyz__ = 5",
    optionD: "None",
    correctOption: "optionD"
},

{
    question: "Is Python case sensitive when dealing with identifiers?",
    optionA: "No",
    optionB: "Yes",
    optionC: "machine dependent",
    optionD: "None",
    correctOption: "optionA"
},

{
    question: "Which of the following is the correct extension of the Python
file?",
    optionA: ".python",
    optionB: ".py",

```

```
    optionC: ".p",
    optionD: ".pyt",
    correctOption: "optionB"
},

{
    question: "Which keyword is used for function in Python language?",
    optionA: "Function",
    optionB: "Fun",
    optionC: "Define",
    optionD: "Def",
    correctOption: "optionD"
},

{
    question: "What does pip stand for python?",
    optionA: "Preferred Installer Program",
    optionB: "unlimited length",
    optionC: "all private members must have leading and trailing underscores",
    optionD: "Program Installer Package",
    correctOption: "optionA"
},

{
    question: "In which language is Python written?",
    optionA: "English",
    optionB: "Html",
    optionC: "C",
    optionD: "R",
    correctOption: "optionC"
}
```

```

    },

    {
        question: "Which of the following functions is a built-in function in python?",
        optionA: "print()",
        optionB: "sqrt()",
        optionC: "scan()",
        optionD: "seed()",
        correctOption: "optionA"
    },

    {
        question: "Los Angeles is also known as ?",
        optionA: "Angels City",
        optionB: "Shining city",
        optionC: "City of Angels",
        optionD: "Lost Angels",
        correctOption: "optionC"
    },

    {
        question: "Which of the following is not a core data type in Python programming?",
        optionA: "Tuples",
        optionB: "Lists",
        optionC: "Dictionary",
        optionD: "Class",
        correctOption: "optionD"
    },

```

```
{
  question: "How many sides does an hexagon have ?",
  optionA: "Six",
  optionB: "Sevene",
  optionC: "Four",
  optionD: "Five",
  correctOption: "optionA"
},

{
  question: "How many planets are currently in the solar system ?",
  optionA: "Eleven",
  optionB: "Seven",
  optionC: "Nine",
  optionD: "Eight",
  correctOption: "optionD"
},

{
  question: "Which Planet is the hottest ?",
  optionA: "Jupitar",
  optionB: "Mercury",
  optionC: "Earth",
  optionD: "Venus",
  correctOption: "optionB"
},

{
  question: "where is the smallest bone in human body located?",
  optionA: "Toes",
```

```
    optionB: "Ears",  
    optionC: "Fingers",  
    optionD: "Nose",  
    correctOption: "optionB"  
  },
```

```
{  
  question: "How many teeth does an adult human have ?",  
  optionA: "28",  
  optionB: "30",  
  optionC: "32",  
  optionD: "36",  
  correctOption: "optionC"  
},
```

```
{  
  question: "How many players are allowed in a cricket match ?",  
  optionA: "10 players",  
  optionB: "11 players",  
  optionC: "9 players",  
  optionD: "12 players",  
  correctOption: "optionB"  
},
```

```
{  
  question: "30 days has _____?",  
  optionA: "January",  
  optionB: "December",
```

```
    optionC: "June",  
    optionD: "August",  
    correctOption: "optionC"  
  },
```

```
{  
  question: "How many hours can be found in a day ?",  
  optionA: "30 hours",  
  optionB: "38 hours",  
  optionC: "48 hours",  
  optionD: "24 hours",  
  correctOption: "optionD"  
},
```

```
{  
  question: "_____ is the hottest Continent on Earth ?",  
  optionA: "Oceania",  
  optionB: "Antarctica",  
  optionC: "Africa",  
  optionD: "North America",  
  correctOption: "optionC"  
},
```

```
{  
  question: "Which country is the largest in the world ?",  
  optionA: "Russia",  
  optionB: "Canada",  
  optionC: "Africa",  
  optionD: "Egypt",
```

```
    correctOption: "optionA"
  },

  {
    question: "Which of these numbers is an odd number ?",
    optionA: "Ten",
    optionB: "Twelve",
    optionC: "Eight",
    optionD: "Eleven",
    correctOption: "optionD"
  },

  {
    question: "Where is the world tallest building located ?",
    optionA: "Africa",
    optionB: "California",
    optionC: "Dubai",
    optionD: "Italy",
    correctOption: "optionC"
  },

  {
    question: "What is the maximum possible length of an identifier?",
    optionA: "15",
    optionB: "14",
    optionC: "None of these",
    optionD: "16",
    correctOption: "optionC"
  },
```

```

    {
      question: "In which year was the Python language developed?",
      optionA: "1991",
      optionB: "1990",
      optionC: "1989",
      optionD: "1992",
      correctOption: "optionC"
    },
  ],

  let shuffledQuestions = [] //empty array to hold shuffled selected questions

  function handleQuestions() {
    //function to shuffle and push 10 questions to shuffledQuestions array
    while (shuffledQuestions.length <= 29) {
      const random = questions[Math.floor(Math.random() * questions.length)]
      if (!shuffledQuestions.includes(random)) {
        shuffledQuestions.push(random)
      }
    }
  }
}

```

```

let questionNumber = 1
let playerScore = 0
let wrongAttempt = 0
let indexNumber = 0

```

```

// function for displaying next question in the array to dom

```

```

function NextQuestion(index) {
  handleQuestions()
  const currentQuestion = shuffledQuestions[index]
  document.getElementById("question-number").innerHTML = questionNumber
  document.getElementById("player-score").innerHTML = playerScore
  document.getElementById("display-question").innerHTML =
currentQuestion.question;
  document.getElementById("option-one-label").innerHTML =
currentQuestion.optionA;
  document.getElementById("option-two-label").innerHTML =
currentQuestion.optionB;
  document.getElementById("option-three-label").innerHTML =
currentQuestion.optionC;
  document.getElementById("option-four-label").innerHTML =
currentQuestion.optionD;

}

```

```

function checkForAnswer() {
  const currentQuestion = shuffledQuestions[indexNumber] //gets current
Question
  const currentQuestionAnswer = currentQuestion.correctOption //gets current
Question's answer
  const options = document.getElementsByName("option"); //gets all elements
in dom with name of 'option' (in this the radio inputs)
  let correctOption = null

  options.forEach((option) => {
    if (option.value === currentQuestionAnswer) {

```

```

        //get's correct's radio input with correct answer
        correctOption = option.labels[0].id
    }
})

//checking to make sure a radio input has been checked or an option being
chosen
if (options[0].checked === false && options[1].checked === false &&
options[2].checked === false && options[3].checked == false) {
    document.getElementById('option-modal').style.display = "flex"
}

//checking if checked radio button is same as answer
options.forEach((option) => {
    if (option.checked === true && option.value === currentQuestionAnswer) {
        document.getElementById(correctOption).style.backgroundColor =
"green"
        playerScore++
        indexNumber++
        //set to delay question number till when next question loads
        setTimeout(() => {
            questionNumber++
        }, 1000)
    }

    else if (option.checked && option.value !== currentQuestionAnswer) {
        const wrongLabelId = option.labels[0].id
        document.getElementById(wrongLabelId).style.backgroundColor = "red"
        document.getElementById(correctOption).style.backgroundColor =

```

```

        wrongAttempt++
        indexNumber++
        //set to delay question number till when next question loads
        setTimeout(() => {
            questionNumber++
        }, 1000)
    }
})
}

```

```

//called when the next button is called
function handleNextQuestion() {
    checkForAnswer()
    unCheckRadioButtons()
    //delays next question displaying for a second
    setTimeout(() => {
        if (indexNumber <= 29) {
            NextQuestion(indexNumber)
        }
        else {
            handleEndGame()
        }
        resetOptionBackground()
    }, 1000);
}

```

```

//sets options background back to null after display the right/wrong colors
function resetOptionBackground() {

```

```

const options = document.getElementsByName("option");
options.forEach((option) => {
    document.getElementById(option.labels[0].id).style.backgroundColor = ""
})
}

```

// unchecking all radio buttons for next question(can be done with map or  
foreach loop also)

```

function unCheckRadioButtons() {
    const options = document.getElementsByName("option");
    for (let i = 0; i < options.length; i++) {
        options[i].checked = false;
    }
}

```

// function for when all questions being answered

```

function handleEndGame() {
    let remark = null
    let remarkColor = null

    // condition check for player remark and remark color
    if (playerScore <= 10) {
        remark = "Bad Grades, Keep Practicing."
        remarkColor = "red"
    }
    else if (playerScore >= 11 && playerScore < 20) {
        remark = "Average Grades, You can do better."
        remarkColor = "orange"
    }
    else if (playerScore >= 21) {

```

```

        remark = "Excellent, Keep the good work going."
        remarkColor = "green"
    }
    const playerGrade = (playerScore / 30) * 100

    //data to display to score board
    document.getElementById('remarks').innerHTML = remark
    document.getElementById('remarks').style.color = remarkColor
    document.getElementById('grade-percentage').innerHTML = playerGrade
    document.getElementById('wrong-answers').innerHTML = wrongAttempt
    document.getElementById('right-answers').innerHTML = playerScore
    document.getElementById('score-modal').style.display = "flex"

}

//closes score modal and resets game
function closeScoreModal() {
    questionNumber = 1
    playerScore = 0
    wrongAttempt = 0
    indexNumber = 0
    shuffledQuestions = []
    NextQuestion(indexNumber)
    document.getElementById('score-modal').style.display = "none"
}

//function to close warning modal
function closeOptionModal() {
    document.getElementById('option-modal').style.display = "none"
}

```